Uniform Black-Box Separations via Non-Malleable Extractors

Marshall Ball[1][0000−0002−4236−3710]⋆ and Dana Dachman-Soled[2][0000−0001−6797−641X]⋆⋆

[1] New York University
[2] University of Maryland

Abstract. We construct t-non-malleable extractors—which allow an attacker to tamper with a source t times—for high min-entropy sources samplable by poly-time hierarchy circuits and for tampering classes corresponding to poly-time hierarchy functions from derandomization-type assumptions.
We then show an application of this new object to ruling out constructions of succinct, non-interactive, arguments (SNARGs) secure against *uniform* adversaries from *uniform* falsifiable assumptions via a class of black-box reductions that has not been previously considered in the literature. This class of black-box reductions allows the reduction to arbitrarily set the *coins*, as well as the input, of the uniform adversary it interacts with. The class of reductions we consider is restricted in allowing only non-adaptive queries to the adversary.

1 Introduction

(Seedless) randomness extractors are deterministic functions that transform an adversarial, unpredictable source into something statistically close to uniformly random bits. An n-bit source, X, is said to have *min-entropy* k if for all x, $\Pr[X = x] \leq 2^{-k}$. Regrettably, deterministic or seedless randomness extractors do not exist for arbitrary n-bit sources, even those with min-entropy $n-1$. However, such randomness extractors *do* exist for more structured source classes: sources formed by concatenating shorter, independent, unpredictable sources [15]; sources samplable by small circuits [53]; etc.

(Seedless) non-malleable extractors, introduced by Cherangchi and Guruswami [14], are randomness extractors with a very strong property that resembles limited independence. Namely, the output of the randomness extractor should be statistically-close to uniform, even if one sees the extractor evaluated on strings related to the source.

In particular, we say a deterministic function NMExt is a (relaxed, seedless) *t-non-malleable extractor* for a class of sources, $\mathcal{X}$, and a class of tampering functions, $\mathcal{F}$, if for any source $X \in \mathcal{X}$ and any sequence of t functions $f_1, \ldots, f_t \in \mathcal{F}$ without fixed points[3], the output

$$[(\text{NMExt}(X), \text{NMExt}(f_1(X)), \ldots, \text{NMExt}(f_t(X)) \approx (\mathcal{U}, \text{NMExt}(f_1(X)), \ldots, \text{NMExt}(f_t(X)).$$

Since their introduction, there has been a concerted effort in constructing *explicit* (seedless) non-malleable extractors for various combinations of source classes and tampering classes. Central to this study has been *non-malleable two-source extractors*: non-malleable extractors for the class of sources that are the concatenation of two independent high min-entropy sources, (X, Y), and the class of tampering functions that may independently tamper each half of the source independently, $(X, Y) \mapsto (f(X), g(Y))$ for any f, g.

Part of the reason for this focus is that non-malleable two-source extractors have found a variety of powerful applications:

⋆ Supported in part by NSF grant CCF-2443735 and the Simons Foundation.
⋆⋆ Supported in part by NSF grants CNS-2154705 and CNS-1933033.

[3] The general notion of a (seedless) non-malleable extractor does not require the fixed point restriction, but is more complicated to state. (Note that if $f_i(X) = X$ then the output of the extractor will be the same. This is resolved in the usual manner, by defining a simulator that outputs unrelated strings and locations where the tampered extractor outputs are the same as the source. The two notions are equivalent up to a small loss in parameters for many natural source/tampering classes.

1. Generic-compilers for *tamper and leakage resilient key-derivation* when an attacker, perhaps due to physical constraints, can only independently leak and tamper on each half of a key at rest.
2. *Split-state non-malleable codes* [14]: means of *encoding* information such that tampering independently with the left and right halves of the codeword either does not change the encoded information or decodes something completely unrelated.
3. *Two-source extractors* [33, 39], particularly in the low min-entropy regime. Perhaps most surprisingly, non-malleable two-source extractors with low error have been central to recent breakthroughs in explicitly constructing two-source extractors (and hence other downstream applications such as Ramsey graphs).
4. *Non-malleable extractors* [7] for other classes of sources and tampering functions.

This last application is perhaps most relevant to the kind of non-malleable extractor studied in this work: non-malleable extractors for *bounded polynomial-size post-selecting sources and tampering functions.*

A size s *post-selecting source*, introduced by Ball et al. [8], is a generalization of size s samplable sources [53] (sources samplable by a size s circuit with uniformly random input bits) and size s recognizable sources [49] (sources uniform over the witness set of a size s circuit). A post-selecting source X is associated with (at least one) size s circuit C with two outputs, an n-bit output $x = C_1(r)$ and a special flag $\phi = C_2(r)$, such that the source is sampled by the circuit on a random input *conditioned* on the flag being set to 1:

$$X \equiv (C_1(\mathcal{U})|C_2(U) = 1).$$

Perhaps most importantly, these sources capture samplable sources after conditioning on efficiently computable leakage on the source.[4] Given this, such extractors naturally have applications in tamper and leakage-resilient key-derivation against a large natural class of sources, leakage, and tampering (where all 3 correspond polynomial time procedures).

The primary focus of this work is yet another surprising application of explicit non-malleable extractors: the impossibility of certain kinds of *uniform black-box reductions.*

By uniform BB reductions we mean the following: The underlying primitive is assumed to be secure against uniform adversaries and the constructed primitive is proven secure against uniform adversaries. The reduction used in the proof of security is itself uniform. The reduction interacts with the uniform adversary and uses it to break the underlying primitive.

In this work, we further consider "generalized black-box reductions," where the reduction is allowed to arbitrarily fix the random tape of the adversary. Previous impossibility results for uniform black-box reductions did not allow the reduction to access the random tape of the adversary. The difficulty in our setting lies in the fact that the uniform adversary (with a constant size description) has *no other source of randomness* other than the arbitrarily set random tape provided by the reduction.

The Case of SNARGs. Succinct non-interactive arguments (SNARGs) are proof systems for languagues in NP with proofs π whose lengths are much smaller than the witness w. Although SNARGs are widely used in practice [47], Gentry and Wichs [22] showed that, under a natural assumption, it is impossible to prove the adaptive soundness of a SNARG via a reduction from a *falsifiable assumption* [42] when the reduction treats the adversary as a black-box. More precisely, they showed that either a black-box reduction that does not access the coins of the adversary does not exist, or that the underlying falsifiable assumption can be broken by a poly-size non-uniform adversary.

The Gentry-Wichs Meta-Reduction. We briefly recap the Gentry-Wichs impossibility proof. Assuming the existence of a polynomial-time, black-box reduction R, they show the existence of an inefficient SNARG adversary A that, given an input crs, breaks soundness by producing proofs π_{no} that verify correctly for statements $x_{no} \notin L$. By assumption, the reduction R with black-box access to A must break the falsifiable assumption when interacting with this adversary. Next, they show the existence of an efficient simulator Sim for A that fools any polynomial-time distinguisher. This implies that R^{Sim} must also break the falsifiable

[4] It should be noted this class also captures samplable sources that are conditioned on leakage on the *intermediate* values of the sampling process.

assumption (otherwise one can distinguish between A and Sim). Finally, since R^{Sim} is an efficient algorithm that breaks the falsifiable assumption, it implies that if a black-box reduction exists, the falsifiable assumption can be broken by poly-time adversaries.

To obtain the SNARG adversary A that breaks soundness, Gentry and Wichs prove a "leakage lemma," stating that if distributions $\mathcal{L}$ and $\bar{\mathcal{L}}$ (over L and $\bar{L}$ resp.) are indistinguishable, then for short leakage π_{yes}, there exists a distribution over leakage π_{no} such that $\{(x_{yes}, \pi_{yes}) : x_{yes} \leftarrow \mathcal{L}\}$ is indistinguishable from $\{(x_{no}, \pi_{no}) : x_{no} \leftarrow \bar{\mathcal{L}}\}$. Gentry-Wichs instantiate their leakage lemma as follows: $\mathcal{L}$ is a distribution on yes-instances of $\mathcal{L}$, $\bar{\mathcal{L}}$ is a distribution on no-instances, and π is the SNARG proof for x. The lemma then yields, for every CRS crs, an augmented no-distribution $(x_{no}, \pi_{no}) \leftarrow \bar{\mathcal{L}}^*_{\mathsf{crs}}$ that is indistinguishable from the SNARG yes-distribution $(x_{yes}, \pi_{yes} \leftarrow \mathcal{L}^*_{\mathsf{crs}}$. Thus, $A(\mathsf{crs})$ is defined to output samples from $\bar{\mathcal{L}}^*_{\mathsf{crs}}$ while $\mathsf{Sim}(\mathsf{crs})$ outputs samples from $\mathcal{L}^*_{\mathsf{crs}}$. Follow-up work of Jetchev and Pietrzak [30] gave a constructive proof of the leakage lemma, bounding the size of the randomized circuit needed to generate the fake proof π_{no}, given the sample $x_{no} \sim \bar{\mathcal{L}}$.

The original Gentry/Wichs proof assumed that the reduction does not access the random coins of the adversary and therefore, upon submitting a crs, only sees random draws from the $\mathcal{L}^*_{\mathsf{crs}}$ or $\bar{\mathcal{L}}^*_{\mathsf{crs}}$ distribution. They did not give a constructive $\bar{\mathcal{L}}^*_{\mathsf{crs}}$ distribution, but just showed its existence. Further, their final simulator (for the non-security parameter preserving case) is non-uniform, since it holds an internal table with draws from the $\bar{\mathcal{L}}^*_{\mathsf{crs}}$ distribution for all security parameters below some threshold. In follow-up work, Vadhan and Zheng [54] used the uniform min-max theorem to remove these last two restrictions: They gave a uniform and constructive $\bar{\mathcal{L}}^*_{\mathsf{crs}}$ distribution and showed, using this, that the real adversary and the simulated adversary could both be made uniform. However, they still assumed that the reduction does not access the random coins of the adversary. In another work of Chung et al. [16], it was shown that one could remove the restriction that the reduction does not access the coins of the adversary by considering a non-uniform, deterministic, adversary that applies an internally hard-wired random oracle to the submitted crs to obtain the random coins for sampling the distributions. Our goal, on the other hand, is to construct a uniform adversary and a uniform simulator, in order to show that even if attempting to construct a SNARG that is secure against uniform adversaries then either (1) no black-box reduction exists (even one that can arbitrarily set the random coins of the adversary) or (2) the underlying assumption is insecure even against *uniform* poly-time adversaries.

1.1 Our Results

Before we begin describing our results, we first make some brief remarks about our assumptions. All our results follow from the existence of strong circuit lower bounds of the form "$\mathsf{E} = \mathsf{DTIME}(2^{O(n)})$ requires exponential-size X-circuits" for some $X \in \{\text{nondeterministic}, \Sigma_2\}$. These types arose in the derandomization literature. Impagliazzo and Wigderson [28] famously showed that "E requires exponential size circuits" implies $\mathsf{BPP} = \mathsf{P}$. Impagliazzo and Wigderson's assumption effectively says that non-uniformity does not always significantly speed up computation. Strengthenings of this assumption to exponential size nondeterministic circuits (circuits that take witnesses in addition to their input) or Σ_i-circuits (circuits with gates that compute a Σ_i-complete function) and related classes were first introduced in the context of derandomizing nondeterminstic complexity classes (e.g. showing sufficient conditions for $\mathsf{AM} = \mathsf{NP}$), and have since seen extensive use in the context of derandomization [9, 18, 21, 25, 27, 32, 41, 50–53].

Improved non-malleable extractors for efficient sources and tampering. Prior work by Ball, Dachman-Soled, and Loss [7] constructed non-malleable extractors for sources samplable or recognizable by polynomial-size circuits and tampered by polynomial-size circuits under a circuit lower bound assumption from the derandomization community. We improve the state of the art for non-malleable extractors in the setting of efficiently samplable sources and efficiently computable sources in 3 dimensions:

1. Many tamperings: our extractors are resilient to t-tamperings where t is slightly less than the min-entropy of the source.[5] Prior work only considered the special case of $t = 1$.

[5] Min-entropy is always an upper-bound on the number of tamperings. Let $E : \{0,1\}^n \to \{0,1\}$ be any function and fix any strings c_0, c_1 such that $E(c_0) = 0$ and $E(c_1) = 1$. Consider X uniform on the first k bits and 0 elsewhere.

2. Wider source class: our extractors work for post-selecting sources. This is a superset of the class of samplable and recognizable sources considered by prior work.
3. Improved assumptions: our extractors are secure under the assumption that $\mathsf{E} = \mathsf{DTIME}(2^{O(n)})$ is hard for exponential-size nondeterministic circuits. Prior work assumed that E was hard for exponential-size Σ_2-circuits.[6] Comparable to the polynomial hierarchy, it is consistent with current knowledge that functions computable by exponential size nondeterministic circuits form a strict subset of those functions computed by exponential size Σ_2-circuits. This assumption matches the best assumptions currently known for such extractors *without any non-malleability property* [8].

Theorem 1 (Informal). *Let $s(n) = \mathsf{poly}(n)$. Assuming $\mathsf{E} = \mathsf{DTIME}(2^{O(n)})$ requires exponential-size nondeterministic circuits, there exists a constant c and an efficiently computable $\tilde{O}(n)$-non-malleable extractor with error $n^{-\Omega(1)}$, $\mathsf{NMExt}_s : \{0,1\}^n \to \{0,1\}^m$ where $m = \tilde{\Omega}(n)$, for the class of size $s(n)$ post-selecting sources on n-bits with min-entropy $k \geq c \cdot n$ and the class of size $s(n)$-tampering functions.*

Moreover, this result relativizes: for any oracle O, if E is hard for exponential-size nondeterministic O-oracle circuits, the above holds with respect to the class of sources samplable by post-selecting size $s(n)$ O-oracle circuits and size $s(n)$ O-oracle tampering functions.

A downside of this result is $n^{-\Omega(1)}$ statistical error. Unfortunately, as shown by Applebaum et al. [3], this is inherent in any randomness extractor built from these assumptions in a black-box way. While this does not suffice for many cryptographic applications, it is sufficient for the flagship application of this paper.

Application: New impossibility results for SNARGs. We consider reductions R that are "black-box" in the adversary–i.e. the reduction accesses the adversary via oracle queries. Further, we consider non-adaptive reductions in which all oracle queries are made simultaneously. Importantly, our oracle allows the reduction R to select the coins of A arbitrarily. In particular, it does not need to sample them uniformly at random.

However, we do restrict the reduction in the following ways:

- There is a fixed constant c such that on input security parameter λ, R only queries A on security parameters $\lambda' \geq \lambda^{1/c}$.
- The number of queries that the reduction makes, $t = \mathsf{poly}(\lambda)$, is a fixed polynomial in the security parameter

Following Pass [46], we consider both restrictions above to fall under the definition of "security-preserving" reductions.

We therefore call the class of reductions that we consider *security-preserving, non-adaptive query, generalized black-box reductions* or *SPNA-generalized black-box reductions.*

We prove the following theorem:

Theorem 2 (Informal). *Assume that an NP language $\mathcal{L}$ has a sub-exponentially hard subset-membership problem and that E requires exponential-size Σ_2-circuits. Let Π be a candidate SNARG for $\mathcal{L}$. Then, for any uniform, falsifiable assumption one of the following must hold:*

- *There is a uniform, poly-time adversary that breaks the falsifiable assumption.*
- *There is no SPNA-generalized black-box reduction showing the uniform soundness of Π based on the falsifiable assumption.*

We believe that our techniques extend to other cases where meta-reductions have been used to rule out constructions of non-interactive primitives from falsifiable assumptions, such as the impossibility result of Pass for perfect non-interactive zero knowledge (NIZK) [45]. A main remaining open question is whether our results can be extended to rule out *adaptive*, generalized black-box reductions as well.

Consider the tampering functions $f_i : x \mapsto c_{x_i}$. Finally, notice that X (and hence $E(X)$) is fully determined by $E(f_1(X)), \ldots, E(f_k(X))$.

[6] Ball et al.'s stated assumptions require hardness against either Σ_3 or Σ_4 circuits, depending on the source class. But hardness against Σ_2-circuits is a corollary of their technical lemmas and subsequent improvements in "vanilla" randomness extractors for these source classes [8].

1.2 Related Work

Non-malleable extractors. There is a long line of research on non-malleable extractors (NME) [17, 14] resilient against various classes of tampering [34–38, 13, 7], with particular attention focused on the case of two-source non-malleable extractors and split-state tampering [34, 37, 38]. There has also been significant work on variants of NME—and the closely related primitive of non-malleable codes [19]—that extend the definitions to various forms of many-time tampering [12, 20, 44, 2].

Seedless extractors for samplable and recognizable sources. Trevisan and Vadhan [53] considered seedless extractors for the class of distributions samplable by bounded polynomial sized circuits. Under the assumption that E requires exponential size Σ_4 circuits, they presented constructions of seedless extractors for linear min-entropy, samplable sources over n bits, that output $\Omega(n)$ bits that are $1/\mathsf{poly}$-close to uniform. Applebaum et al. [3] showed that the $1/\mathsf{poly}$ error is somewhat inherent by ruling out black-box reductions in this setting. They introduced a notion of *relative-error* extractors and showed that if the output of the extractor is $1/\mathsf{poly}$-close to uniform with relative error, then every event occurs w.r.t. the output distribution with probability at most $(1 + 1/\mathsf{poly})$ times the probability it occurs w.r.t. the uniform distribution. In particular, events that are negligible under the uniform distributions cannot become noticeable under the distribution outputted by the extractor. Under the assumption that E requires exponential size Σ_4 circuits, they constructed relative-error seedless extractors whose outputs are $1/\mathsf{poly}$-close to uniform with relative error for linear min-entropy, samplable sources. Under the assumption that E requires exponential size Σ_3 circuits, they constructed relative-error seedless extractors whose outputs are $1/\mathsf{poly}$-close to uniform with relative error for linear min-entropy, recognizable sources.

The meta reduction technique has been used in various works to rule out (among others) the construction of non-interactive primitives—such as perfect non-interactive zero-knowledge (NIZK)—from all falsifiable assumptions [45, 11]. To the best of our knowledge, all of these results construct an inefficient adversary that samples responses from some distribution and they do not allow the reduction to access the random coins of the adversary. As mentioned above, Chung et al. [16] extended the technique to remove this restriction, but at the cost of hardwiring a random oracle into the inefficient adversary, and then considering the behavior of the reduction relative to a draw from a distribution over non-uniform adversaries.

A recent line of work has managed to almost fully circumvent the Gentry-Wichs impossibility result and obtain adaptive SNARGs (for NP and UP) from falsifiable assumptions[56–58, 40]. The reason these results do not contradict the Gentry-Wichs impossibility is that their notion of adaptivity is slightly weaker. Specifically, in the recent results, the circuit computing the NP-relation, and hence the length of the statement itself, must be fixed in advance, before the crs is published. Then, the adversary breaking soundness can choose a statement that depends on the crs, but whose length is upper-bounded by the circuit's input length. On the other hand, in the Gentry-Wichs impossibility result, it is crucial for the construction of the attacker that the length of the crs is fixed *first*, then an NP-language that is at least $2^{|\mathsf{crs}|}$-hard, and a corresponding statement, is chosen.

In the case of security-parameter preserving reductions for SNARGs, Campanelli et al. [11] ruled out even the weaker form of adaptive soundness described above (where the crs size can grow with the length of the statement size). However, that result also does not rule out reductions that access the random coins of the adversary. Thus, one implication of our result is extending their impossibility result even to the case where the reduction accesses the random coins of the adversary but is non-adaptive.

2 Technical Overview

Before we begin explaining our results, we need to introduce a critical ingredient in both constructions: *seed-extending pseudorandom generators* [31]. A seed-extending PRG, introduced by Kinne, Van Melkebeek, and Shaltiel, for a class $\mathcal{C}$ is a PRG whose output contains the seed as a prefix, $G(s) = (s, y)$ for some y, and fools any $C \in \mathcal{C}$. Observe that "cryptographic" seed-extending PRGs, by which we mean fixed polytime-computable and PPT-secure seed-extending PRGs, are impossible: one can always distinguish by evaluating

G. Thus, such PRGs are only possible in the "complexity-theoretic" or "derandomization" regime where G is not computable in the adversarial class $\mathcal{C}$. Kinne et al. [31] observed that the standard presentation of Nisan and Wigderson's generator [43] is indeed seed-extending and hence follows from standard derandomization assumptions, such as those used in this work.

2.1 Our t-NMExt Construction

We begin by recapping the NMExt constructon of Ball et al. [7] for single-time tampering for the class of bounded polynomial-size circuits. Our goal is to extend this construction to t-time tampering (and to tampering circuits in higher complexity classes). The NMExt of [7] is simple: On input x, the non-malleable extractor applies a deterministic extractor Ext for samplable sources to obtain σ, then applies a seed extending PRG to σ to obtain $G(\sigma) = (\sigma||y)$, and finally applies a two-source non-malleable extractor $\mathsf{2NMExt}((\sigma||y), x)$ to obtain the final output.

Our t-NMExt construction is nearly identical, except we replace the two-source non-malleable extractor with a two-source t-non-malleable extractor. We first recall the security proof of Ball et al. and then point out the key differences.

The security proof for the non-malleable extractor proceeds by a proof by contradiction: Assume NMExt does not achieve non-malleability against tampering functions $f \in \mathcal{F}$. Using the definition of relaxed non-malleability, this means that, for some $f \in \mathcal{F}$, the statistical distance between the distributions $\{(\mathsf{NMExt}(X), \mathsf{NMExt}(f(X))\}$ and $\{(U, \mathsf{NMExt}(f(X))\}$ is at least $1/\mathsf{poly}$. Now, we use this fact to construct a constant-round private-coin interactive proof for proving (roughly) that $G(\sigma) = (\sigma||y)$, from which we will derive a contradiction to the security of PRG G against non-deterministic distinguishers. This follows by applying classical results to convert the IP to a nondeterministic circuit via (a) emulating with a public coin protocol [26], (b) collapsing the rounds of result to an AM protocol [6], and (c) derandomizing the AM protocol with non-uniformity [1]. So in short, it suffices to construct a constant-round IP.

In the IP protocol, both parties receive as input (σ, y), where y is either equal to $G(\sigma)$ or is chosen uniformly at random and independently of σ.

Arthur pre-samples x such that $\mathsf{Ext}(x) = \sigma$. Then Arthur computes $\tilde{\sigma} = \mathsf{Ext}(f(x))$ and sends it to Merlin. Merlin is supposed to compute $(\tilde{\sigma}||\tilde{y}) = G(\tilde{\sigma})$ and send $\tilde{y}$ back to Arthur. Arthur then computes $\tilde{z} = \mathsf{2NMExt}((\tilde{\sigma}, \tilde{y}), \tilde{x})$ and computes z to be either $\mathsf{2NMExt}((\sigma, y), z)$ or to be uniform random. Arthur sends $(z, \tilde{z})$ to Merlin and asks Merlin to guess whether z was chosen as the output of 2NMExt or uniformly at random. If Merlin guesses correctly, Arthur guesses that $(\sigma||y) = G(\sigma)$. Otherwise, Arthur guesses that y is uniform random.

Completeness of the IP protocol follows immediately from the assumption that NMExt does not achieve relaxed non-malleability, and hence the two distributions are distinguishable by Merlin.

To argue soundness, one can show that due to the security of the 2NMExt, regardless of the $\tilde{y}$ returned by (a dishonest) Merlin, the output of the 2NMExt is statistically close to uniform. They argue this by showing that the 2NMExt is actually also a *strong* 2NMExt–i.e. indistinguishability of the distributions holds even given one of the inputs to the 2NMExt. In our case, Merlin knows the entire (σ, y) since it is part of the input. However, the $\tilde{y}$ that is returned by Merlin is *independent of x, conditioned on $\sigma = \mathsf{Ext}(x)$, and* $\tilde{\sigma} = \mathsf{Ext}(f(x))$. Thus, by making the length of x sufficiently longer than the sum of the lengths of σ and $\tilde{\sigma}$, one can ensure that x still has sufficiently high min-entropy conditioned on $\sigma, \tilde{\sigma}$. Therefore, due to the security guarantee of the 2NMExt, Merlin cannot distinguish the two distributions with high probability.

In the t-tampering setting, Arthur simply computes all t tampered seeds $\tilde{\sigma}_1, \ldots, \tilde{\sigma}_t$ and sends all of them over to Merlin to expand. Note that any malicious Merlin can now be recast as a t-tampering strategy on the left source. The rest of the proof proceeds similarly.

One missing piece from the proof sketch above is how Arthur can efficiently sample the source conditioned on the output of extractor being correct (or even sample a post-selecting source at all, given that output is only produced when the flag $\phi = 1$). Previously, this was accomplished by equipping Arthur with an NP-oracle and running classical uniform witness sampling algorithms. However, equipping Arthur with such an oracle ultimately yields a Σ_2-circuit breaking the PRG.

Here, we instead observe that a careful inspection of Goldwasser and Sipser's public coin emulation [26] (see also parametric improvement and modern presentation by Goldreich and Leshkowitz [23]) reveals that only a significant (multiplicative) gap between the completeness and soundness parameters is required to convert an IP to an AM protocol. This is because this emulation simply proves an (approximate) bound on the number of accepting coins that Arthur could have. Critically, this means that it is not necessary that Arthur ever actually accept with non-negligible probability when designing an IP! Instead we can design a non-functional IP with a completeness/soundness gap and simply compile the result to get a functional public coin protocol.

Concretely, in our IP we have Arthur simply give up and reject whenever the sampling fails, which almost always happens. However, because we are conditioning on the output of an *extractor* taking a certain value (and recall that an extractor's output is statistically close to uniform), we can (with a little work) get fine-grained bounds on the exponentially-small probability of successfully sampling and, thus, ultimately control the relative gap between completeness and soundness.

Prior work constructed non-malleable extractors for samplable sources assuming E is hard for exponential-size Σ_2-circuits and samplable source extractors. The upshot of this new analytical approach is that to construct non-malleable extractors for post-selecting sources one need not invoke any additional assumption beyond those known to yield extractors for post-selecting sources: E is hard for exponential-size *nondeterminisitic* circuits.

Finally, we note that both this proof and Ball et al.'s extractor result [8] relativize. So, by giving Arthur access to an oracle O we can "lift" this technique to handle sources sampled via post-selecting O-oracle circuits[7] of polynomial size and O-oracle tampering circuits of polynomial size, at the expense of assuming E is hard for exponential-size nondeterministic O-oracle circuits.

2.2 Application: Ruling Out Falsifiable Reductions to SNARGs

As a first attempt, we consider an adversary A that depends on the reduction R and which responds with $\perp$ if the query it is given, (crs, r), occurs too frequently with respect to the output distribution of R. (Specifically, conditioned on (crs, r) *not* having the above property, the marginal distribution produced by the reduction interacting with the external challenger will have high min-entropy). Since r is uniformly distributed in the real security game, and since there is a relatively small number of such pairs (crs, r), A will refuse to respond with only negligible probability in the real security game, and thus will still be a legitimate SNARG adversary. On the other hand, if the individual query does not occur too frequently, A will apply a seedless, deterministic extractor[8] to the query (since conditioned on this event, the marginal distribution on the query has sufficiently high min-entropy) to obtain v, and then a seed-extending PRG to v to obtain $(v||w)$. Finally, the adversary will use the w outputted by the PRG to sample (x_{no}, π_{no}) from the augmented "no" distribution $\bar{\mathcal{L}}^*_{\mathsf{crs}}$ discussed above, and will return (x_{no}, π_{no}). The goal, as in Gentry-Wichs, is to switch to an efficient adversary, who computes w but then returns (x_{yes}, π_{yes}) sampled from the augmented "yes" distribution $\mathcal{L}^*_{\mathsf{crs}}$ discussed above.

There are three problems with this initial approach:

- The biggest problem is that the adversary we design must be *stateless*, so it can only extract entropy from an individual query (crs, r) in order to sample (x_{no}, π_{no}), as sketched above. On the other hand, the reduction's view includes the entire set of queries and responses, and the reduction may choose queries that are correlated and inter-dependent. However, if the queries are correlated, the randomness w extracted from the queries may also be correlated. It is therefore not clear how we can argue that the reduction cannot distinguish the joint distribution of correlated draws from $\bar{\mathcal{L}}^*_{\mathsf{crs}}$ from correlated draws from $\mathcal{L}^*_{\mathsf{crs}}$. For example, when using correlated randomness to sample, we cannot rule out a case where the correlated draws from the augmented "yes" distribution, $\mathcal{L}^*_{\mathsf{crs}}$, reveal the witness, whereas in the case of the augmented "no" distribution, $\bar{\mathcal{L}}^*_{\mathsf{crs}}$, no witness exists.

[7] These are circuits with O-gates in addition to the standard basis, $\{\wedge, \vee, \neg\}$.
[8] See for example, [53, 8].

– The second main problem is that in the meta-reduction proof strategy, we must ultimately switch to the efficient adversary, Sim, without the reduction detecting it. However, an efficient Sim cannot check whether a submitted query (crs, r) occurs frequently or not, since checking this requires (approximately) counting the number of pre-images of a given output. This means that the simulator must be indistinguishable from the real adversary, while also behaving identically in the case that an indiviudal query occurs frequently or not.
– The final problem arises in the proof of indistinguishability of the real adversary and Sim: When we switch from the true output of the seed-extending PRG to a random output, we still need to be able to sample from the augmented "no" distribution, $\bar{\mathcal{L}}^*_{\mathsf{crs}}$, in order to simulate the hybrid distribution. This sampling process is computationally bounded, but not poly-time. This means that we require the seed-extending PRG to be hard even against distinguishers who can sample from $\bar{\mathcal{L}}^*_{\mathsf{crs}}$. But since sampling from $\bar{\mathcal{L}}^*_{\mathsf{crs}}$ is inefficient, such a seed-extending PRG cannot be poly-time computable.

To solve the first problem, we use $(t-1)$-*non-malleable extractors* against NP-tampering functions. The main observation is that, since the reduction is non-adaptive, for any $i \in [t]$, and $j \neq i$, we can view the j-th query (crs_j, r_j) as the output of an NP-tampering function applied to the i-th output (crs_i, r_i). More precisely, given input (crs_i, r_i), the j-th tampering function uses its NP powers to pre-sample randomness U for the reduction/challenger (see Theorem 5), and runs the reduction/challenger forwards on U to compute the j-th query.

To solve the second problem, in the case where an individual query occurs frequently, the *real* adversary will actually sample (x_{yes}, π_{yes}) from $\mathcal{L}^*_{\mathsf{crs}}$, instead of outputting $\perp$. Ultimately, we will switch from an inefficient adversary that samples from $\bar{\mathcal{L}}^*_{\mathsf{crs}}$ in the case of low frequency, to a poly-time simulator that always samples from $\mathcal{L}^*_{\mathsf{crs}}$. Thus, the simulator ultimately does not have to check whether a query occurs frequently or not, since it will always apply a poly-time computable PRG (which is hard for the reduction to compute, but still poly-time), and then use the randomness to sample from $\mathcal{L}^*_{\mathsf{crs}}$.

To resolve the third problem, we modify the real adversary to use *two* different PRG's: G_{hard} and G_{easy}. G_{hard} is hard even for distinguishers who can sample from $\bar{\mathcal{L}}^*_{\mathsf{crs}}$, whereas G_{easy} is only hard for bounded poly-time distinguishers (such as the reduction itself). G_{hard} will be applied in the case that an individual query has low frequency, whereas G_{easy} will be applied in the case that an individual query has high frequency. When we switch from the inefficient adversary who samples from $\bar{\mathcal{L}}^*_{\mathsf{crs}}$ to the efficient adversary who samples from $\mathcal{L}^*_{\mathsf{crs}}$, we will also switch the PRG from G_{hard} to G_{easy}, using a careful sequence of hybrids. Thus, the final simulator always does the following, regardless of the frequency of the individual query (crs, r) (1) applies the $(t-1)$-non-malleable extractor to obtain v; (2) applies the seed-extending PRG G_{easy} to v to obtain $(v||w)$; (3) uses the resulting random coins w to sample (x_{yes}, π_{yes}) from $\mathcal{L}^*_{\mathsf{crs}}$ and (4) outputs (x_{yes}, π_{yes}) to the reduction. Note that the final simulator is now a uniform, polynomial time algorithm.

3 Preliminaries

3.1 Notation

For a distribution $\mathcal{D}$, we denote by $\mathcal{D}(u)$ a draw from $\mathcal{D}$ using coins u.

We denote by U_k a uniform random variable over k bits. When the number of bits, k, is clear from context or implicit, we use U for simplified notation.

For $S \subseteq N$, where $S = \{i_1, \ldots, i_\ell : i_1 < \cdots < i_\ell\}$ and any n-ary string of values $x_1, \ldots, x_n$, let x_S denote the string $(x_{i_1}, \ldots, x_{i_\ell})$. For random variables X, Y, we write $\Delta(X;Y) \leq \epsilon$ or $X \approx_\epsilon Y$ if the total variation distance between their distributions is at most ϵ.

3.2 Complexity classes and assumptions

We take E to denote $\mathsf{DTIME}[2^{O(n)}]$ the class of languages decidable by deterministic Turing machines in 2^{cn}-time for some constant c. We take circuits to denote circuits over the standard basis $\{\vee, \wedge, \neg\}$. For any language O, an O-oracle aided circuit is a circuit that has special gates that decide O, in addition to the

standard-basis. For any circuit, we say it has size s if it contains at most s gates. We say it has depth d if the longest path from any input to any output gate is of size d. A circuit family, $\{C_n\}_{n\in\mathbb{N}}$, is a collection of circuits such that C_n takes inputs of length n. We take the $\mathsf{SIZE}[s(n)]$ to denote the function families computable by a circuit family $\{C_n\}_{n\in\mathbb{N}}$ such that C_n has size at most $s(n)$, for large enough n. Similarly, we take $\mathsf{SIZE}^O[s(n)]$ to denote the function families computable by an O-oracle aided circuit family $\{C_n\}_{n\in\mathbb{N}}$ such that C_n has size at most $s(n)$, for sufficiently large n.

3.3 Seedless t-non-malleable extractors

Definition 1 (Tampering functions). *For any $n > 0$, let $\mathcal{H}_n$ denote the set of all functions $h : \{0,1\}^n \to \{0,1\}^n$. Any subset $\mathcal{G} \subseteq \mathcal{H}_n$ is a family of tampering functions. For any class of boolean functions $\mathcal{F} = \{f : \{0,1\}^n \to \{0,1\}\}$, we take $\mathcal{F}^n$ to denote the class of n-output functions where each output is computed by some function in $\mathcal{F}$, i.e. $\mathcal{F}^n = \{f_{i_1,\ldots,i_n} : x \mapsto f_{i_1}(x), \ldots, f_{i_n}(x) \mid f_{i_1}, \ldots, f_{i_n} \in \mathcal{F}\}$.*

The particular classes of tampering functions we consider in this work:

- Tampering where each output is computable by an $s(n)$-size circuit, $\mathsf{SIZE}[s(n)]$.
- Tampering where each output is computable by an $s(n)$-size circuit with O-oracle gates, $\mathsf{SIZE}^O[s(n)]$.
- Split-state tampering where two halves of an input are tampered independently and arbitrarily: $\{(\tau_L, \tau_R) : x_1, \ldots, x_{2n} \mapsto \tau_L(x_1, \ldots, x_n), \tau_R(x_{n+1}, \ldots, x_{2n}) | \tau_L, \tau_R \in \mathcal{H}_n\}$.

Definition 2 (Relaxed Seedless t-non-malleable extractor). *Let $\mathcal{X}$ be a family of sources on $\{0,1\}^n$ and $\mathcal{F}$ be a class of tampering functions acting on $\{0,1\}^n$. Further assume that all $f \in \mathcal{F}$ does not have any fixed points. A function* $\mathrm{NMExt} : \{0,1\}^n \to \{0,1\}^m$ *is defined to be a relaxed (ϵ, t)-non-malleable extractor with respect to $\mathcal{X}$ and $\mathcal{F}$ if the following hold: for any $X \in \mathcal{X}$ and any set $\{f_1, \ldots, f_t\}$, such that for all $i \in [t]$, $f_i \in \mathcal{F}$, we have*

$$\Delta\Big((\mathrm{NMExt}(X), \mathrm{NMExt}(f_1(X)), \ldots, \mathrm{NMExt}(f_t(X)));$$
$$(\mathcal{U}_m, \mathrm{NMExt}(f_1(X)), \ldots, \mathrm{NMExt}(f_t(X))\Big) \leq \epsilon.$$

3.4 Seed-extending pseudorandom generators

Definition 3 ([31]). *A function $G : \{0,1\}^\ell \to \{0,1\}^n$ is said to be an* ϵ-pseudorandom generator (PRG) *for a class $\mathcal{C}$, if for all $C \in \mathcal{C}$,*

$$\Delta(C(G(\mathcal{U}_\ell)); C(\mathcal{U}_n)) \leq \epsilon$$

A PRG, G, is said to be seed-extending *if the prefix of its output is its input, i.e. $G(\sigma) = \sigma, G'(\sigma)$ for some function $G' : \{0,1\}^\ell \to \{0,1\}^{n-\ell}$.*

We are principally concerned with seed-extending PRGs against various types of circuits of a given size: non-deterministic circuits, non-deterministic NP-circuits, etc. Throughout this paper, we take a PRG for a class of circuits of size s to mean a $1/s$-PRG for that class of circuits. Note that because we are interested in both seed-extending PRGs, as well as PRGs for non-deterministic circuits, so-called "cryptographic" PRGs which can be easily evaluated by the classes they are constructed to fool do not suffice: a distinguisher given the seed, or nondeterminism, can easily determine if a string is in the PRG's image. Thankfully, as observed by Kinne et al. [31], Nisan and Wigderson's seminal construction yields a seed extending PRG, provided one starts with an appropriately hard function. We conclude with the formal theorem statement.

Theorem 3 ([31, 28, 32, 50, 51, 3]). *If E requires exponential size circuits of type $X \in \{\textit{deterministic}, \textit{nondeterministic}, \mathsf{NP}, \Sigma_i\}$, then for every constant $c > 1$ there exists a constant $\alpha > 1$ such that for every sufficiently large n, and every r such that $\alpha \log n \leq \ell \leq n$ there is a seed-extending PRG, $\mathsf{G} : \{0,1\}^\ell \to \{0,1\}^n$, for size n^c circuits of type $X \in \{\textit{deterministic}, \textit{nondeterministic}, \mathsf{NP}, \Sigma_i\}$.*

3.5 Witness Sampling

We also require the following classical, relativizing results on approximate counting and sampling NP-witnesses.

Theorem 4 (Approximate Counting with an NP-oracle [29]). *For every $i \geq 0$, every sufficiently large s and every $\epsilon > 0$, there is a Σ_{i+1}-circuit A of size $\mathsf{poly}(s/\epsilon)$ that given a Σ_i-circuit $C : \{0,1\}^n \to \{0,1\}$ of size s outputs a value $\widehat{M}$ such that*

$$\widehat{M} \in (1 \pm \epsilon)|\{x : C(x) = 1\}|$$

Theorem 5 (Sampling Witnesses with an NP-oracle [10, 29]). *For every $i \geq 0$, every sufficiently large s and every $\delta > 0$, there is a randomized Σ_{i+1}-circuit A of size $\mathsf{poly}(s/\log(1/\delta))$ that given a Σ_i-circuit $C : \{0,1\}^n \to \{0,1\}$ of size s outputs a value in $\{0,1\}^n \cup \perp$ such that for every size s Σ_i-circuit, $\Pr[A(C) = \perp] \leq \delta$ and the distribution $(A(C)|A(C) \neq \perp)$ is uniform over $\{x : C(x) = 1\}$.*

3.6 SNARGs and Black-Box Reductions

Our work is concerned with provable security for SNARGs, defined below. Note that we require *adaptive* soundness, i.e., soundness even against adversaries that are allowed to select their instance after seeing the common reference string.

Definition 4 (Uniform Succinct Non-Interactive Arguments for NP). *A SNARG system $\Pi = (\mathsf{G}, \mathsf{P}, \mathsf{V})$:*

- $(\mathsf{crs}, \mathsf{priv}) \leftarrow \mathsf{G}(1^\lambda)$*: The generation algorithm takes as input the security parameter λ and outputs a common reference string* crs *as well as private verification information* priv*.*
- $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$*: The prove algorithm takes in the* crs*, a statement x, and a witness w, and outputs a proof π.*
- $b \leftarrow \mathsf{V}(\mathsf{priv}, x, \pi)$*: the verification algorithm takes as input* priv*, a statement x, and a proof π, and outputs a bit b (1 denotes acceptance, 0 denotes rejection).*

We say that Π is a succinct non-interactive argument (SNARG) for a language L in $\mathcal{NP}$ with corresponding relation $\mathcal{R}$ if it satisfies the following three properties:

Completeness: *For all $(x, w) \in \mathcal{R}$,*

$$\Pr\left[\mathsf{V}(\mathsf{priv}, x, \pi) = 0 \,\middle|\, \begin{array}{c}(\mathsf{crs}, \mathsf{priv}) \leftarrow \mathsf{G}(1^\lambda) \\ \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)\end{array}\right] = \mathsf{negl}(\lambda)\,. \tag{1}$$

Uniform Adaptive Soundness: *For all uniform, PPT $\bar{\mathsf{P}}$,*

$$\Pr\left[\mathsf{V}(\mathsf{priv}, x, \pi) = 1 \wedge x \notin L \,\middle|\, \begin{array}{c}(\mathsf{crs}, \mathsf{priv}) \leftarrow \mathsf{G}(1^\lambda) \\ (x, \pi) \leftarrow \bar{\mathsf{P}}(\mathsf{crs})\end{array}\right] = \mathsf{negl}(\lambda)\,. \tag{2}$$

Succinctness: *All proofs π produced by P satisfy $|\pi| = \mathsf{poly}(\lambda)(|x| + |w|)^{o(1)}$.*

We say that a PPT $\bar{\mathsf{P}}$ breaks a SNARG system $\Pi = (\mathsf{G}, \mathsf{P}, \mathsf{V})$ if the probability that V outputs 1 in the soundness experiment with $\bar{\mathsf{P}}$ exceeds $1/p(\lambda)$ for some polynomial p and infinitely many λ.

Definition 5 (Uniform Falsifiable Assumption). *A uniform falsifiable assumption is a pair (M, c) where M is an interactive uniform poly-time algorithm ("the challenger") and c is a constant ("the guessing probability"). The output of M is a single bit; we let the output 1 mean* **win**.

Such an assumption (M, c) is said to be true if, for every uniform, PPT A,

$$\Pr[\langle \mathsf{M}, A\rangle(1^\lambda) = \mathbf{win}] \leq c + \mathsf{negl}(\lambda)\,, \tag{3}$$

where the probability is taken over the coins of M and A.

We say that a uniform, PPT A breaks the assumption (M, c) if the probability that it wins the security experiment $\langle \mathsf{M}, A \rangle$ exceeds $c + 1/p(\lambda)$ for some polynomial p and infinitely many λ.

Definition 6 (Non-Adaptive Black-Box Reduction for SNARGs). *Let Π be a SNARG and (M, c) a falsifiable assumption. A non-adaptive, black-box reduction (establishing the soundness of Π, based on the assumption (M, c)) is an interactive PPT oracle algorithm R such that for every uniform adversary A that breaks the SNARG system Π, R^A breaks the assumption (M, c). Further, all of R's quries to A must be submitted simultaneously.*

Note that the oracle in the above definition allows the reduction R to select the coins of A in any manner at all. In particular, it need not sample them uniformly at random.

However we do restrict the reduction in the following ways:

- There is a fixed constant c such that on input security parameter λ, R only queries A on security parameters $\lambda' \geq \lambda^{1/c}$.
- The number of queries the reduction makes, $t = \mathsf{poly}(\lambda)$, is a fixed polynomial in the security parameter (and in particular is independent of the length of the random tape of the adversary).

Following Pass [46], we consider both restrictions above to fall under the definition of "security-preserving" reductions.

We call the class of non-adaptive, black-box reductions that can arbitrarily set the adversary's random tape, but have the above two restrictions *security-preserving, non-adaptive query, generalized black-box reductions* or *SPNA-generalized black-box reductions.*

4 Main Application

In this section, we use our construction of $(t-1)$-non-malleable extractors as a building block to obtain the following result:

Theorem 6. *Assume that an NP language $\mathcal{L}$ has a sub-exponentially hard subset-membership problem and that E requires exponential-size Σ_2-circuits. Let $\Pi = (\mathsf{G}, \mathsf{P}, \mathsf{V})$ be a candidate SNARG for $\mathcal{L}$, satisfying the completeness and succinctness properties. Then, for any uniform, falsifiable assumption (M, c), one of the following must hold:*

- *There is a uniform, poly-time adversary that breaks (M, c).*
- *There is no SPNA-generalized black-box reduction showing the uniform soundness of Π based on the assumption (M, c).*

Towards proving the theorem, we assume the existence of a SPNA-generalized black-box reduction R making $t := t(\lambda)$ queries on security parameter λ. See Figure 1 for a pictorial representation. We next describe certain building-blocks that will be used to define our SNARG adversary A.

4.1 Hard Languages and Leakage Simulation Theorem

We assume existence of an NP-language L and distributions $\mathcal{L}^+, \bar{\mathcal{L}}$ where $\mathcal{L}^+$ is a distribution over statement, witness pairs (x, w), where in $x \in L$, and $\bar{\mathcal{L}}$ is a distribution over no instances. The distribution over the first output of $\mathcal{L}^+$ is denoted as $\mathcal{L}$. We require that $\mathcal{L}$ and $\bar{\mathcal{L}}$ should be indistinguishable against non-uniform adversaries of size $\mathsf{poly}(2^{\lambda_{pi}})$.

For a fixed crs, let $\mathcal{L}^*_{\mathsf{crs}}$ denote the joint distribution (x, π) of statements $x \sim \mathcal{L}$ and corresponding proofs π with respect to crs. We can efficiently sample from $\mathcal{L}^*_{\mathsf{crs}}$ by sampling $(x, w) \sim \mathcal{L}^+$ and honestly computing the proof π using w w.r.t crs. Assume the length of π is λ_π.

We have the following theorem by Jetchev and Pietrzak [30]:

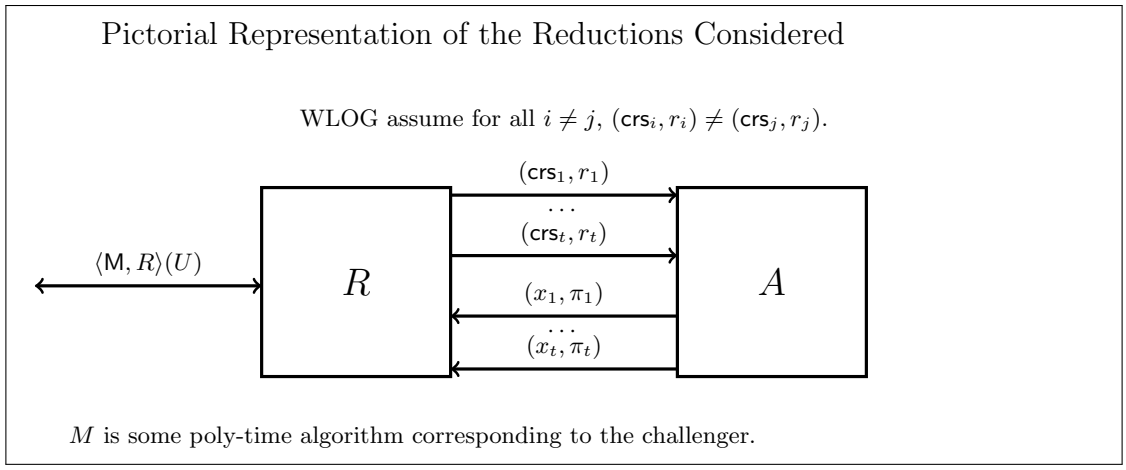

Fig. 1. A SPNA-generalized black-box reduction from a uniform, falsifiable assumption to uniform SNARGs.

Theorem 7 ([30]). *For every* crs*, there exists a* $\mathsf{poly}(2^{\lambda_\pi})$*-size randomized circuit* C_{crs} *such that for all* $\mathsf{poly}(\lambda)$*-sized distinguishers* D

$$\left| \Pr_{(x,\pi)\sim\mathcal{L}^*_{\mathsf{crs}}}[D(x,\pi)=1] - \Pr_{x\sim\mathcal{L},\rho\sim\{0,1\}^{p(\lambda)}}[D(x, C_{\mathsf{crs}}(x;\rho))=1] \right| \leq \frac{1}{\mathsf{poly}(\lambda)}.$$

We now define the distribution used by the SNARG adversary to sample false statements and proofs. For every fixed crs, let $\bar{\mathcal{L}}^*_{\mathsf{crs}}$ be the following distribution:

- Sample $x \sim \bar{\mathcal{L}}$
- Sample $\pi \sim C_{\mathsf{crs}}(x;U)$, where U is a uniform random variable, and C_{crs} is the lexicographically first randomized circuit of size $\mathsf{poly}(2^{\lambda_\pi})$ guaranteed by Theorem 7.

Using the hardness of distinguishing $\mathcal{L}$ and $\bar{\mathcal{L}}$ we obtain the following corollary:

Corollary 1. *For every* crs *and for all* $\mathsf{poly}(\lambda)$*-sized distinguishers* D*,*

$$\left| \Pr_{(x,\pi)\sim\mathcal{L}^*_{\mathsf{crs}}}[D(x,\pi)=1] - \Pr_{(x,\pi)\sim\bar{\mathcal{L}}^*_{\mathsf{crs}}}[D(x,\pi)=1] \right| \leq \frac{1}{\mathsf{poly}(\lambda)}.$$

4.2 $(t-1)$-Non-Malleable Extractor

We require a relaxed $(t-1, 1/\mathsf{poly})$-non-malleable extractor NMExt that is computable in uniform polynomial-time for high-min-entropy sources Z sampled by bounded $\mathsf{poly}(\lambda)$-size Σ_2-circuits. NMExt must be secure against the class $\mathcal{F}$ of tampering functions f consisting of bounded $\mathsf{poly}(\lambda)$-size NP-circuits. We show in Section 5.2 that such non-malleable extractors can be constructed under the assumption that E requires exponential-size Σ_2-circuits.

4.3 Seed-Extending PRG's

We need two types of seed-extending PRG's:

- G_{hard} takes as input a seed of length security parameter λ. G_{hard} can take even time exponential in λ to compute, but it must be *uniform*. G_{hard} is secure against bounded $\mathsf{poly}(\lambda)$-size Σ_2-circuits and is also secure against $\mathsf{poly}(2^{\lambda_\pi})$-size (regular) circuits.
- G_{easy} takes as input a seed of length security parameter λ. G_{easy} must be computable in *uniform* polynomial-time. G_{easy} must be secure against bounded $\mathsf{poly}(\lambda)$-size Σ_2-circuits .

Theorem 3 implies that such seed-extending PRG's can be constructed under the assumption that E requires exponential-size Σ_2-circuits.

4.4 Distributions we consider

We next define several distributions and the tampering functions that will be used inf the definition of the real adversary A, as well as in the Hybrid argument used in our proof, where we switch from the inefficient adversary A to an efficient simulator.

Distribution over R's queries: Let $\mathcal{D}$ be the following poly-samplable distribution:
On input $u \sim U$:

1. Run $\langle \mathsf{M}, R \rangle(u)$ to obtain $(\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)$.
2. Output $(\overrightarrow{\mathsf{crs}}, \vec{r}) := (\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)$.

For $i \in [t]$, we denote by $\mathcal{D}[i][1]$ the output crs_i.
For $i \in [t]$, let $\mathcal{D}_i$ be the distribution over the i-th (crs_i, r_i) pair outputted by $\mathcal{D}$. We further let $\mathcal{D}_i[1]$ or $\mathcal{D}_i(u)[1]$ indicate the crs part of the output, whereas $\mathcal{D}_i[2]$ or $\mathcal{D}_i(u)[2]$ indicate the randomness part of the output.

Set of "infrequent" queries made by R. Let $\mathcal{S}$ be the set

$$\left\{ r : \forall i \in [t], \Pr_{\mathcal{D}_i[2]}[r] \leq \frac{2^{\log^2(\lambda)}}{2^{|r|}} \right\}.$$

Probability a query is "infrequent." For $i \in [t]$, let p_i be the probability $\Pr_{r_i \sim \mathcal{D}_i[2]}[r_i \in \mathcal{S}]$.

Distributions over R's queries, conditioned on the query being "infrequent."

- For $i \in [t]$, let Z_i be the distribution $(\mathsf{crs}_i, r_i) \sim \mathcal{D}_i \mid (r_i \in \mathcal{S})$.
- For $i \in [t]$, let C_i be the distribution $(\overrightarrow{\mathsf{crs}}, \vec{r}) = ((\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)) \sim \mathcal{D} | r_i \in \mathcal{S}$.

Distribution over i-th Reduction query, conditioned on all queried crs's. For $i \in [t]$, let $Z_i(\overrightarrow{\mathsf{crs}})$ be the distribution $\mathcal{D}_i(u)$, where u is chosen uniformly at random from the set $\{u : \mathcal{D}_i(u)[2] \in \mathcal{S} \wedge \mathcal{D}_1(u)[1] = \mathsf{crs}_1 \wedge \cdots \wedge \mathcal{D}_t(u)[1] = \mathsf{crs}_t)\}$.

Note that

- If $p_i \geq \frac{1}{2^{\log^2(\lambda)}}$ then with all but negligible probability over choice of $\overrightarrow{\mathsf{crs}} \sim C_i$, the min entropy of $Z_i(\overrightarrow{\mathsf{crs}})$ is at least $|r| - t|\mathsf{crs}| - 3\log^2(\lambda)$. This is a sufficiently high fraction for extraction w.r.t. $|\mathsf{crs}| + |r|$ when $|r| \gg t|\mathsf{crs}|$.
- At most $\frac{t \cdot 2^{|r|}}{2^{\log^2(\lambda)}}$ number of random tapes do not fall into the set $\mathcal{S}$. Therefore

$$\Pr_{r \sim U}[r \notin \mathcal{S}] \leq \mathsf{negl}(\lambda).$$

The second bullet will imply that our real adversary A is a valid SNARG adversary even though it does not break the soundness of the SNARG (it will return a proof of a true statement) when $r \notin \mathcal{S}$.

Complexity of sampling. As written above, sampling from $Z_i(\overrightarrow{\mathsf{crs}})$ requires access to a $\#\mathsf{P}$ oracle, since checking membership of $r \in \mathcal{S}$ requires computing the exact number of pre-images u such that $\mathcal{D}_i(u)[2] = r$. We chose to define $\mathcal{S}$ in this way for conceptual simplicity. In the following, we describe an alternate definition of $\mathcal{S}$ that allows for more efficient sampling.

Existence of a *relative error approximator* that can be deterministically computed using a uniform NP circuit follows from an assumption we already require (that E requires exponential-size non-deterministic circuits) [48]. On input r, the relative error approximator outputs a value $\rho(r)$ such that for all r,

$$(1 - \epsilon) \cdot \Pr_{r \sim \mathcal{D}_i[2]}[r] \leq \rho(r) \leq \Pr_{r \sim \mathcal{D}_i[2]}[r].$$

We therefore re-define the set $\mathcal{S} := \{r : \rho(r) \leq (1-\epsilon) \cdot \frac{2^{\log^2(\lambda)}}{2^{|r|}}\}$. Note that if $r \in \mathcal{S}$ then $\Pr_{r\sim\mathcal{D}_i[2]}[r] \leq \frac{2^{\log^2(\lambda)}}{2^{|r|}}$. This means that the min-entropy conditioned on being in $\mathcal{S}$ remains the same as in our previous definition of $\mathcal{S}$. On the other hand, if $r \notin \mathcal{S}$, then $\Pr_{r\sim\mathcal{D}_i[2]}[r] \geq (1-\epsilon)\frac{2^{\log^2(\lambda)}}{2^{|r|}}$. This implies that the fraction of random tapes that do not fall into the set $\mathcal{S}$ will still be negligibly small and so A is still a valid SNARG adversary. Further, using this alternate definition of $\mathcal{S}$, the distribution $Z_i(\overrightarrow{crs})$ itself can be sampled by Σ_2-circuits.

4.5 Tampering functions

For $i, j \in [t]$, fixed $\overrightarrow{\mathsf{crs}}$, and fixed random coins ρ, let $f^i_{j,\overrightarrow{\mathsf{crs}},\rho}$ be the tampering function that uses ρ to sample a pre-image u uniformly at random from the set: $\{u : \mathcal{D}_i(u)[2] = r_i \wedge \mathcal{D}_1(u)[1] = \mathsf{crs}_1 \wedge \cdots \wedge \mathcal{D}_t(u)[1] = \mathsf{crs}_t\}$. Then $f^i_{j,\overrightarrow{\mathsf{crs}},\rho}$ outputs $D_j(u)$. Such a tampering function can be implemented using a polynomial-size NP circuit. See Theorem 5 for more details.

Note that our tampering functions have *no fixed points* since the reduction never outputs the same query twice (in such a case the reduction could simply respond to its own query by copying the previous output).

Since $Z_i(\overrightarrow{\mathsf{crs}})$ has high min-entropy with all but negligible probability over choice of $\overrightarrow{\mathsf{crs}} \sim C_i$, by relaxed $(t-1)$-non-malleability, we have that for $i \in [t]$, and for every ρ, with all but negligible probability over $\overrightarrow{\mathsf{crs}} \sim C_i$:

$$\begin{aligned}&\left(\mathsf{NMExt}(Z_i(\overrightarrow{\mathsf{crs}})), [\mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(Z_i(\overrightarrow{\mathsf{crs}})))]_{j\in[t]\setminus\{i\}}\right)\\ &\quad\overset{s}{\approx} \left(U, [\mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(Z_i(\overrightarrow{\mathsf{crs}})))]_{j\in[t]\setminus\{i\}}\right). \end{aligned} \tag{4}$$

4.6 The real adversary A

We now define the real, inefficient, uniform SNARG adversary A:

The real adversary A. On input (crs, r):

1. Using brute-force search, find the lexicographically first circuit C_{crs} of size $\mathsf{poly}(2^{\lambda_\pi})$ guaranteed by Theorem 7.
2. Compute $v = \mathsf{NMExt}(\mathsf{crs}, r)$.
3. If $(\mathsf{crs}, r) \in \mathcal{S}$
   (a) Compute $(v, w) = G_{hard}(v)$. Note that G_{hard} takes superpolynomial time to compute. Let $w = w^1||w^2$.
   (b) Set $x = \bar{\mathcal{L}}(w^1)$
   (c) Set $\pi = C_{\mathsf{crs}}(x; w^2)$.
   (d) Output (x, π)
4. If $(\mathsf{crs}, r) \notin \mathcal{S}$
   (a) Compute $(v, w) = G_{easy}(v)$. Note that G_{easy} takes polynomial time to compute.
   (b) Compute $(x, \pi) = \mathcal{L}^*_{\mathsf{crs}}(w)$ in polynomial time.
   (c) Output (x, π).

We consider the augmented view of $\langle \mathsf{M}, R\rangle$ that includes the entire random string u that the challenger M and reduction R use during their interaction (this allows full reconstruction of the joint view of R and the challenger corresponding to the falsifiable assumption.). In particular, the views of both R and M can be fully recovered given the augmented view.

$$\mathsf{View}^{Real}_{\mathsf{aug}} = (u, (x_1, \pi_1), (x_2, \pi_2)).$$

4.7 The uniform PPT Simulator

The simulator Sim. On input (crs, r):

1. Compute $v = \mathsf{NMExt}(\mathsf{crs}, r)$.
2. Compute $(v, w) = G_{easy}(v)$. Note that G_{easy} takes polynomial time to compute.
3. Compute $(x, \pi) = \mathcal{L}^*_{\mathsf{crs}}(w)$ in polynomial time.
4. Output (x, π).

4.8 Hybrid argument for t parallel queries

We will now argue–via a sequence of hybrids–that the real, inefficient, adversary A can be simulated by the uniform, PPT simulator Sim given in the previous section. Specifically, the real experiment generates a distribution over augmented views that is identical to the distribution in Hybrid $H_{1,0}$, whereas the simulator generates a distribution over augmented views that is identical to the distribution in Hybrid $H_{t,5}$. We will argue that each pair of consecutive hybrids is either perfectly, statistically, or computationally indistinguishable with distinguishing probability $1/\mathsf{poly}$ for some poly. Since it can be checked in polynomial time that the augmented view of the real experiment consists of a "break" of the underlying falsifiable assumption, it must therefore also be the case that the efficiently generated simulated distribution consists of a "break" of the underlying falsifiable assumption with $1/\mathsf{poly}$ probability.

For $i \in [1, \ldots, t]$, we define the following sequence of hybrids: $H_{1,0}, H_{1,1}, \ldots, H_{1,5}, H_{2,0}, H_{2,1}, \ldots, H_{t,5}$.

Hybrid $H_{i,0}$. In this Hybrid, we switch the order of sampling. Specifically, letting p_i be the probability that the i-th query is not a "frequent query," with probability p_i, we first sample the t values of the crs, we then sample the t outputs of NMExt and the derived coins of the adversary, we then *pre-sample* the randomness u of the reduction and challenger and re-compute the entire augmented view, where the responses of the adversary for the first $i-1$ queries are chosen from the "yes" distribution, whereas the responses for the last $t-i+1$ queries are chosen from the "no" distribution. With probability $1-p_i$, we do not change the distribution of this Hybird, and we sample the view by running experiment $H_{i-1,5}$, conditioned on the i-th query being a "frequent query." The formal description of the hybrid follows:

Let $p_i := \Pr_{r_i \sim \mathcal{D}_i[2]}[r_i \in \mathcal{S}]$. With probability p_i do the following:

1. Sample u uniformly at random from the set $\{u : \mathcal{D}_i(u)[2] \in \mathcal{S}\}$. Fix $\mathsf{crs}_i = \mathcal{D}_i(u)[1]$, and fix ρ uniformly at random. For $j \in [t] \setminus \{i\}$, fix $\mathsf{crs}_j = D_j(u)[1]$ and $v_j = \mathsf{NMExt}(f_{\mathsf{crs}_j}(\mathcal{D}_i(u)))$. Note that the distribution over $\overrightarrow{\mathsf{crs}} := (\mathsf{crs}_1, \ldots, \mathsf{crs}_t)$ is identical to a draw from C_1. Further, $\mathcal{D}_i(u)$ conditioned on $\overrightarrow{\mathsf{crs}}$ is identical to the distribution $Z_i(\overrightarrow{\mathsf{crs}})$.
2. For $j \geq i$, fix the circuits C_{crs_j}, each of size $\mathsf{poly}(2^{\lambda_\pi})$.
3. For $j > i$, compute $v_j || w_{j,hard} = G_{hard}(v_j)$. Let $w_{j,hard} = w^1_{j,hard} || w^2_{j,hard}$. Set $x_{j,no} = \bar{\mathcal{L}}(w^1_{j,hard}), \pi_{j,no} = C_{\mathsf{crs}_j}(x_{j,no}; w^2_{j,hard})$.
4. For $j \in [t] \setminus \{i\}$, compute $v_j || w_{j,easy} = G_{easy}(v_j)$, Set $(x_{j,yes}, \pi_{j,yes}) = \mathcal{L}^*_{\mathsf{crs}_j}(w_{j,easy})$.
5. Sample $v_i = \mathsf{NMExt}(Z_i(\overrightarrow{\mathsf{crs}}))$, conditioned on $v_j = \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho})(Z_i(\overrightarrow{\mathsf{crs}})))$ for $j \in [t] \setminus \{i\}$.
6. Compute $v_i || w_i = G_{hard}(v_i)$. Let $w_i = w^1_i || w^2_i$.
7. Pre-sample u uniformly at random from the set

$$\{u : \mathcal{D}_i(u)[2] \in \mathcal{S} \wedge \mathcal{D}_1(u)[1] = \mathsf{crs}_1 \cdots \wedge \mathcal{D}_t(u)[1] = \mathsf{crs}_t \wedge \\ \text{for } j \in [t] \setminus \{i\},\ \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(\mathcal{D}_i(u))) = v_j \wedge \mathsf{NMExt}(\mathcal{D}_i(u)) = v_i\}.$$

8. Using u, $(\overrightarrow{\mathsf{crs}}, \overrightarrow{r}) = (\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)$ can be efficiently computed.
9. For $j < i$, set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
10. For $j > i$, if $(\mathsf{crs}_j, r_j) \in \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,no}, \pi_{j,no})$ and if $(\mathsf{crs}_j, r_j) \notin \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
11. Set $x_i = \bar{\mathcal{L}}(w^1_i)$
12. Set $\pi_i = C_{\mathsf{crs}_i}(x_i; w^2_i)$.

With probability $1-p_i$, run experiment $H_{i-1,5}$, conditioned on $(\mathsf{crs}_i, r_i) \notin \mathcal{S}$.
We denote the augmented view in this hybrid as

$$\mathsf{View}^{H_{i,0}}_{\mathsf{aug}} = (u, (x_1, \pi_1), \ldots, (x_t, \pi_2)).$$

Note that hybrid $H_{1,0}$ is identical to the real experiment the only difference is the order of sampling.

Hybrid $H_{i,1}$. In this Hybrid, we replace the sampled output of the non-malleable extractor on the i-th query (v_i) with a uniform random value. To allow for the switch, we rely on the statistical properties of the $(t-1)$-non-malleable extractor against the distribution $Z_i(\overrightarrow{\mathsf{crs}})$ (which we have argued has min-entropy and is samplable by bounded $\mathsf{poly}(\lambda)$-size Σ_2-circuits in Section 4.4, and the tampering functions $f^i_{j,\overrightarrow{\mathsf{crs}},\rho}$, which we have argued can be computed by $\mathsf{poly}(\lambda)$-size Σ_2-circuits in Section 4.5. The formal description of the hybrid follows:

Let $p_i := \Pr_{r_i \sim \mathcal{D}_i[2]}[r_i \in \mathcal{S}]$. With probability p_i do the following:

1. Sample u uniformly at random from the set $\{u : \mathcal{D}_i(u)[2] \in \mathcal{S}\}$. Fix $\mathsf{crs}_i = \mathcal{D}_i(u)[1]$ and fix ρ at random. For $j \in [t] \setminus \{i\}$, fix $\mathsf{crs}_j = D_j(u)[1]$ and $v_j = \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(\mathcal{D}_i(u)))$. Note that the distribution over $\overrightarrow{\mathsf{crs}} := (\mathsf{crs}_1, \ldots, \mathsf{crs}_t)$ is identical to a draw from C_1. Further, $\mathcal{D}_i(u)$ conditioned on $\overrightarrow{\mathsf{crs}}$ is identical to the distribution $Z_i(\overrightarrow{\mathsf{crs}})$.
2. Fix the circuits $C_{\mathsf{crs}_1}, \ldots, C_{\mathsf{crs}_t}$, each of size $\mathsf{poly}(2^{\lambda_\pi})$.
3. For $j > i$, compute $v_j || w_{j,hard} = G_{hard}(v_j)$. Let $w_{j,hard} = w^1_{j,hard} || w^2_{j,hard}$. Set $x_{j,no} = \bar{\mathcal{L}}(w^1_{j,hard}), \pi_{j,no} = C_{\mathsf{crs}_j}(x_{j,no}; w^2_{j,hard})$.
4. For $j \in [t] \setminus \{i\}$, compute $v_j || w_{j,easy} = G_{easy}(v_j)$, Set $(x_{j,yes}, \pi_{j,yes}) = \mathcal{L}^*_{\mathsf{crs}_j}(w_{j,easy})$.
5. Sample v_i uniformly at random.
6. Compute $v_i || w_i = G_{hard}(v_i)$. Let $w_i = w^1_i || w^2_i$.
7. Pre-sample u uniformly at random from the set

$$\{u : \mathcal{D}_i(u)[2] \in \mathcal{S} \wedge \mathcal{D}_1(u)[1] = \mathsf{crs}_1 \cdots \wedge \mathcal{D}_t(u)[1] = \mathsf{crs}_t \wedge \\ \text{for } j \in [t] \setminus \{i\},\ \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(\mathcal{D}_i(u))) = v_j \wedge \mathsf{NMExt}(\mathcal{D}_i(u)) = v_i\}.$$

8. Using u, $(\overrightarrow{\mathsf{crs}}, \overrightarrow{r}) = (\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)$ can be efficiently computed.
9. For $j < i$, set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
10. For $j > i$, if $(\mathsf{crs}_j, r_j) \in \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,no}, \pi_{j,no})$ and if $(\mathsf{crs}_j, r_j) \notin \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
11. Set $x_i = \bar{\mathcal{L}}(w^1_i)$
12. Set $\pi_i = C_{\mathsf{crs}_i}(x_i; w^2_i)$.

With probability $1 - p_i$, run experiment $H_{i-1,5}$, conditioned on $(\mathsf{crs}_i, r_i) \notin \mathcal{S}$.

Claim.

$$\mathsf{View}^{H_{i,0}}_{\mathsf{aug}} \overset{s}{\approx} \mathsf{View}^{H_{i,1}}_{\mathsf{aug}}$$

Proof. If p_i is negligible, then $H_{i,0}$ and $H_{i,1}$ are both statistically close to the distribution $H_{i-1,5}$, conditioned on $(\mathsf{crs}_i, r_i) \notin \mathcal{S}$ (and hence to each other). On the other hand, if $p_i \geq \frac{1}{2^{\log^2(\lambda)}}$, then the distribution $Z_i(\overrightarrow{\mathsf{crs}})$ has high min-entropy and can be sampled by bounded $\mathsf{poly}(\lambda)$-size Σ_2-circuits (as shown in Section 4.4), and the tampering functions $f^i_{j,\overrightarrow{\mathsf{crs}},\rho}$ can also be computed by $\mathsf{poly}(\lambda)$-size Σ_2-circuits (as shown in Section 4.5). Therefore, due to the relaxed $(t-1)$-non-malleability of NMExt (Def 2), $\mathsf{View}^{H_{i,0}}_{\mathsf{aug}}$ and $\mathsf{View}^{H_{i,1}}_{\mathsf{aug}}$ are $1/\mathsf{poly}$-statistically close.

Hybrid $H_{i,2}$. In this Hybrid, we replace the output of the "hard" PRG $G_{hard}(v_i)$ with a random string w_i. Note that this PRG is sufficiently hard that we can compute the remainder of the hybrid distribution (which includes pre-sampling the coins u of the reduction and challenger and computing the fake proof $\pi_i = C_{\mathsf{crs}_i}(x_i; w^2_i)$) without breaking the security of the PRG (See Section 4.3). Thus, indistinguishability of the hybrids will follow from the security of the PRG. The formal description of the hybrid follows:

Let $p_i := \Pr_{r_i \sim \mathcal{D}_i[2]}[r_i \in \mathcal{S}]$. With probability p_i do the following:

1. Sample u uniformly at random from the set $\{u : \mathcal{D}_i(u)[2] \in \mathcal{S}\}$. Fix $\mathsf{crs}_i = \mathcal{D}_i(u)[1]$ and fix ρ at random. For $j \in [t] \setminus \{i\}$, fix $\mathsf{crs}_j = D_j(u)[1]$ and $v_j = \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(\mathcal{D}_i(u)))$. Note that the distribution over $\overrightarrow{\mathsf{crs}} := (\mathsf{crs}_1, \ldots, \mathsf{crs}_t)$ is identical to a draw from C_1. Further, $\mathcal{D}_i(u)$ conditioned on $\overrightarrow{\mathsf{crs}}$ is identical to the distribution $Z_i(\overrightarrow{\mathsf{crs}})$.

2. Fix the circuits $C_{\mathsf{crs}_1}, \ldots, C_{\mathsf{crs}_t}$, each of size $\mathsf{poly}(2^{\lambda_\pi})$.
3. For $j > i$, compute $v_j||w_{j,hard} = G_{hard}(v_j)$. Let $w_{j,hard} = w^1_{j,hard}||w^2_{j,hard}$. Set $x_{j,no} = \bar{\mathcal{L}}(w^1_{j,hard}), \pi_{j,no} = C_{\mathsf{crs}_j}(x_{j,no}; w^2_{j,hard})$.
4. For $j \in [t] \setminus \{i\}$, compute $v_j||w_{j,easy} = G_{easy}(v_j)$, Set $(x_{j,yes}, \pi_{j,yes}) = \mathcal{L}^*_{\mathsf{crs}_j}(w_{j,easy})$.
5. Sample v_i uniformly at random.
6. Sample w_i uniformly at random. Let $w_i = w^1_i||w^2_i$.
7. Pre-sample u uniformly at random from the set

$$\{u : \mathcal{D}_i(u)[2] \in \mathcal{S} \wedge \mathcal{D}_1(u)[1] = \mathsf{crs}_1 \cdots \wedge \mathcal{D}_t(u)[1] = \mathsf{crs}_t \wedge \\ \text{for } j \in [t] \setminus \{i\}, \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(\mathcal{D}_i(u))) = v_j \wedge \mathsf{NMExt}(\mathcal{D}_i(u)) = v_i\}.$$

8. Using u, $(\overrightarrow{\mathsf{crs}}, \vec{r}) = (\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)$ can be efficiently computed.
9. For $j < i$, set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
10. For $j > i$, if $(\mathsf{crs}_j, r_j) \in \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,no}, \pi_{j,no})$ and if $(\mathsf{crs}_j, r_j) \notin \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
11. Set $x_i = \bar{\mathcal{L}}(w^1_i)$
12. Set $\pi_i = C_{\mathsf{crs}_i}(x_i; w^2_i)$.

With probability $1 - p_i$, run experiment $H_{i-1,5}$, conditioned on $(\mathsf{crs}_i, r_i) \notin \mathcal{S}$.

Claim.

$$\mathsf{View}^{H_{i,1}}_{\mathsf{aug}} \overset{c}{\approx} \mathsf{View}^{H_{i,2}}_{\mathsf{aug}}$$

Proof. Consider the following distribution over circuits:

Non-uniform advice.

1. Sample u uniformly at random from the set $\{u : \mathcal{D}_i(u)[2] \in \mathcal{S}\}$. Fix $\mathsf{crs}_i = \mathcal{D}_i(u)[1]$ and fix ρ at random. For $j \in [t] \setminus \{i\}$, fix $\mathsf{crs}_j = D_j(u)[1]$ and $v_j = \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(\mathcal{D}_i(u)))$.
2. Fix the circuits $C_{\mathsf{crs}_j}, j \geq i$, each of size $\mathsf{poly}(2^{\lambda_\pi})$.
3. For $j \in [t] \setminus \{i\}$, compute $v_j||w_{j,easy} = G_{easy}(v_j)$, Set $(x_{j,yes}, \pi_{j,yes}) = \mathcal{L}^*_{\mathsf{crs}_j}(w_{j,easy})$.
4. For $j > i$, $v_j||w_{j,hard} = G_{hard}(v_j)$. Let $w_{j,hard} = w^1_{j,hard}||w^2_{j,hard}$. Set $x_{j,no} = \bar{\mathcal{L}}(w^1_{j,hard}), \pi_{j,no} = C_{\mathsf{crs}_2}(x_{j,no}; w^2_{j,hard})$.

Given an input $v_i||w_i$, where $w_i = w^1_i||w^2_i$ the circuit does as follows:

1. Pre-sample u uniformly at random from the set

$$\{u : \mathcal{D}_i(u)[2] \in \mathcal{S} \wedge \mathcal{D}_1(u)[1] = \mathsf{crs}_1 \cdots \wedge \mathcal{D}_t(u)[1] = \mathsf{crs}_t \wedge \\ \text{for } j \in [t] \setminus \{i\}, \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(\mathcal{D}_i(u))) = v_j \wedge \mathsf{NMExt}(\mathcal{D}_i(u)) = v_i\}.$$

2. Using u, $(\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)$ can be efficiently computed.
3. For $j < i$, set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
4. For $j > i$, if $(\mathsf{crs}_j, r_j) \in \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,no}, \pi_{j,no})$ and if $(\mathsf{crs}_j, r_j) \notin \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
5. Set $x_i = \bar{\mathcal{L}}(w^1_i)$
6. Set $\pi_i = C_{\mathsf{crs}_i}(x_i; w^2_i)$.

Assume $(v_i||w_i)$ is either random or the output of a PRG $G_{hard}(v_i)$ that is computed in time $2^{\mathsf{poly}(\lambda)}$ and is secure against Σ_2-circuits of polynomial size $p_1(\lambda)$ (needed to pre-sample u, and check whether $(crs_j, r_j) \in \mathcal{S}$), and (regular) circuits of size $\mathsf{poly}(2^{\lambda_\pi})$ (needed to compute the circuit $C_{crs_i}(x_i; w^2_i)$.

If $(v_i||w_i)$ is the output of G_{hard}, then we exactly obtain the Hybrid 1 distribution, whereas if $(v_i||w_i)$ is random, we exactly obtain the Hybrid 2 distribution. By the security of the PRG G_{hard} against this complexity class, Hybrid 1 and Hybrid 2 are indistinguishable.

Hybrid $H_{i,3}$ In this Hybrid, we replace the "no" instance $x_i = \overline{\mathcal{L}}(w_i^1), \pi_i = C_{\mathsf{crs}_i}(x_i; w_i^2)$ with a "yes" instance $(x_i, \pi_i) = \mathcal{L}^*_{\mathsf{crs}_i}(w_i)$. Since the value of w_i is independent of all other random variables in the experiment, we can performs all the inefficient parts of the experiment in a pre-processing stage. In the online stage, all that must be done is plugging in (x_i, π_i) as the i-th response of the oracle and using the pre-sampled random coins u (given as non-uniform advice) to complete the experiment. The formal description of the hybrid follows:

Let $p_i := \Pr_{r_i \sim \mathcal{D}_i[2]}[r_i \in \mathcal{S}]$. With probability p_i do the following:

1. Sample u uniformly at random from the set $\{u : \mathcal{D}_i(u)[2] \in \mathcal{S}\}$. Fix $\mathsf{crs}_i = \mathcal{D}_i(u)[1]$ and fix ρ at random. For $j \in [t] \setminus \{i\}$, fix $\mathsf{crs}_j = D_j(u)[1]$ and $v_j = \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(\mathcal{D}_i(u)))$. Note that the distribution over $\overrightarrow{\mathsf{crs}} := (\mathsf{crs}_1, \ldots, \mathsf{crs}_t)$ is identical to a draw from C_1. Further, $\mathcal{D}_i(u)$ conditioned on $\overrightarrow{\mathsf{crs}}$ is identical to the distribution $Z_i(\overrightarrow{\mathsf{crs}})$.
2. Fix the circuits $C_{\mathsf{crs}_1}, \ldots, C_{\mathsf{crs}_t}$, each of size $\mathsf{poly}(2^{\lambda_\pi})$.
3. For $j > i$, compute $v_j || w_{j,hard} = G_{hard}(v_j)$. Let $w_{j,hard} = w^1_{j,hard} || w^2_{j,hard}$. Set $x_{j,no} = \bar{\mathcal{L}}(w^1_{j,hard}), \pi_{j,no} = C_{\mathsf{crs}_j}(x_{j,no}; w^2_{j,hard})$.
4. For $j \in [t] \setminus \{i\}$, compute $v_j || w_{j,easy} = G_{easy}(v_j)$, Set $(x_{j,yes}, \pi_{j,yes}) = \mathcal{L}^*_{\mathsf{crs}_j}(w_{j,easy})$.
5. Sample v_i uniformly at random.
6. Sample w_i uniformly at random.
7. Pre-sample u uniformly at random from the set

$$\{u : \mathcal{D}_i(u)[2] \in \mathcal{S} \wedge \mathcal{D}_1(u)[1] = \mathsf{crs}_1 \cdots \wedge \mathcal{D}_t(u)[1] = \mathsf{crs}_t \wedge \\ \text{for } j \in [t] \setminus \{i\},\ \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(\mathcal{D}_i(u))) = v_j \wedge \mathsf{NMExt}(\mathcal{D}_i(u)) = v_i\}.$$

8. Using u, $(\overrightarrow{\mathsf{crs}}, \overrightarrow{r}) = (\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)$ can be efficiently computed.
9. For $j < i$, set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
10. For $j > i$, if $(\mathsf{crs}_j, r_j) \in \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,no}, \pi_{j,no})$ and if $(\mathsf{crs}_j, r_j) \notin \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
11. Set $(x_i, \pi_i) = \mathcal{L}^*_{\mathsf{crs}_i}(w_i)$.

With probability $1 - p_i$, run experiment $H_{i-1,5}$, conditioned on $(\mathsf{crs}_i, r_i) \notin \mathcal{S}$.

Claim.

$$\mathsf{View}^{H_{i,2}}_{\mathsf{aug}} \overset{c}{\approx} \mathsf{View}^{H_{i,3}}_{\mathsf{aug}}$$

Proof. Since w_i is independent of all other random variables in the experiment, we will use the non-uniform advice to pre-sample a view missing only (x_i, π_i), which are both computed from $w_i = w_i^1 || w_i^2$.

Specifically, we sample $\left(u, [(x_j, \pi_j)]_{j \in [t] \setminus \{i\}}\right)$ and note that $(\overrightarrow{\mathsf{crs}}, \overrightarrow{r}) = (\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)$ can be computed in polynomial time given u. These are hardwired into the circuit as non-uniform advice.

The circuit then receives (x_i, π_i), where either $(x_i, \pi_i) \sim \mathcal{D}^*_{yes,\mathsf{crs}_i}$ or $(x_i, \pi_i) \sim \mathcal{D}^*_{no,\mathsf{crs}_i}$. The circuit returns the view $(u, (x_1, \pi_1), \ldots, (x_t, \pi_t))$ to the adversary and outputs whatever the adversary does.

Note that if $(x_i, \pi_i) \sim \mathcal{D}^*_{no,\mathsf{crs}_i}$ then the view of the adversary is identical to Hybrid $H_{i,3}$, whereas if $(x_i, \pi_i) \sim \mathcal{D}^*_{yes,\mathsf{crs}_i}$ the view of the adversary is identical to Hybrid $H_{i,4}$. Thus, a distinguishing adversary implies a contradiction to Corollary 1.

Hybrid $H_{i,4}$ In this Hybrid, we switch back from w_i being chosen uniformly at random, to w_i being computed using a PRG. However, this time we use $(v_i || w_i) = G_{easy}$ instead of G_{hard}. Note that at this stage, the only inefficient parts of the hybrid experiment are the pre-sampling of u conditioned on fixing certain random variables in the experiment. Since we chose G_{easy} to be a poly-time computable PRG that is secure against NP-circuits of polynomial size, this pre-sampling step can be done without breaking the security of G_{easy} (see Section 4.3). Thus, indistinguishability of the hybrids will follow from the security of the PRG. The formal description of the hybrid follows:

Let $p_i := \Pr_{r_i \sim \mathcal{D}_i[2]}[r_i \in \mathcal{S}]$. With probability p_i do the following:

1. Sample u uniformly at random from the set $\{u : \mathcal{D}_i(u)[2] \in \mathcal{S}\}$. Fix $\mathsf{crs}_i = \mathcal{D}_i(u)[1]$ and fix ρ at random. For $j \in [t] \setminus \{i\}$, fix $\mathsf{crs}_j = D_j(u)[1]$ and $v_j = \mathsf{NMExt}(f^i_{j,\vec{\mathsf{crs}},\rho}(\mathcal{D}_i(u)))$. Note that the distribution over $\vec{\mathsf{crs}} := (\mathsf{crs}_1, \ldots, \mathsf{crs}_t)$ is identical to a draw from C_1. Further, $\mathcal{D}_i(u)$ conditioned on $\vec{\mathsf{crs}}$ is identical to the distribution $Z_i(\vec{\mathsf{crs}})$.
2. Fix the circuits $C_{\mathsf{crs}_1}, \ldots, C_{\mathsf{crs}_t}$, each of size $\mathsf{poly}(2^{\lambda_\pi})$.
3. For $j > i$, compute $v_j||w_{j,hard} = G_{hard}(v_j)$. Let $w_{j,hard} = w^1_{j,hard}||w^2_{j,hard}$. Set $x_{j,no} = \bar{\mathcal{L}}(w^1_{j,hard}), \pi_{j,no} = C_{\mathsf{crs}_j}(x_{j,no}; w^2_{j,hard})$.
4. For $j \in [t] \setminus \{i\}$, compute $v_j||w_{j,easy} = G_{easy}(v_j)$, Set $(x_{j,yes}, \pi_{j,yes}) = \mathcal{L}^*_{\mathsf{crs}_j}(w_{j,easy})$.
5. Sample v_i uniformly at random.
6. Set $(v_i||w_i) = G_{easy}(v_i)$.
7. Pre-sample u uniformly at random from the set

$$\{u : \mathcal{D}_i(u)[2] \in \mathcal{S} \wedge \mathcal{D}_1(u)[1] = \mathsf{crs}_1 \cdots \wedge \mathcal{D}_t(u)[1] = \mathsf{crs}_t \wedge \\ \text{for } j \in [t] \setminus \{i\}, \mathsf{NMExt}(f^i_{j,\vec{\mathsf{crs}},\rho}(\mathcal{D}_i(u))) = v_j \wedge \mathsf{NMExt}(\mathcal{D}_i(u)) = v_i\}.$$

8. Using u, $(\vec{\mathsf{crs}}, \vec{r}) = (\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)$ can be efficiently computed.
9. For $j < i$, set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
10. For $j > i$, if $(\mathsf{crs}_j, r_j) \in \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,no}, \pi_{j,no})$ and if $(\mathsf{crs}_j, r_j) \notin \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
11. Set $(x_i, \pi_i) = \mathcal{L}^*_{\mathsf{crs}_i}(w_i)$.

With probability $1 - p_i$, run experiment $H_{i-1,5}$, conditioned on $(\mathsf{crs}_i, r_i) \notin \mathcal{S}$.

Claim.

$$\mathsf{View}^{H_{i,3}}_{\mathsf{aug}} \overset{c}{\approx} \mathsf{View}^{H_{i,4}}_{\mathsf{aug}}$$

Proof. Consider the following distribution over circuits:

Non-uniform advice.

1. Sample u uniformly at random from the set $\{u : \mathcal{D}_i(u)[2] \in \mathcal{S}\}$. Fix $\mathsf{crs}_i = \mathcal{D}_i(u)[1]$ and fix ρ at random. For $j \in [t] \setminus \{i\}$, fix $\mathsf{crs}_j = D_j(u)[1]$ and $v_j = \mathsf{NMExt}(f^i_{j,\vec{\mathsf{crs}},\rho}(\mathcal{D}_i(u)))$.
2. Fix the circuits $C_{\mathsf{crs}_j}, j \geq i$, each of size $\mathsf{poly}(2^{\lambda_\pi})$.
3. For $j > i$, compute $v_j||w_{j,hard} = G_{hard}(v_j)$. Let $w_{j,hard} = w^1_{j,hard}||w^2_{j,hard}$. Set $x_{j,no} = \bar{\mathcal{L}}(w^1_{j,hard}), \pi_{j,no} = C_{\mathsf{crs}_j}(x_{j,no}; w^2_{j,hard})$.
4. For $j \in [t] \setminus \{i\}$, compute $v_j||w_{j,easy} = G_{easy}(v_j)$, Set $(x_{j,yes}, \pi_{j,yes}) = \mathcal{L}^*_{\mathsf{crs}_j}(w_{j,easy})$.

Given an input $v_i||w_i$, the circuit does as follows:

1. Pre-sample u uniformly at random from the set

$$\{u : \mathcal{D}_i(u)[2] \in \mathcal{S} \wedge \mathcal{D}_1(u)[1] = \mathsf{crs}_1 \cdots \wedge \mathcal{D}_t(u)[1] = \mathsf{crs}_t \wedge \\ \text{for } j \in [t] \setminus \{i\}, \mathsf{NMExt}(f^i_{j,\vec{\mathsf{crs}},\rho}(\mathcal{D}_i(u))) = v_j \wedge \mathsf{NMExt}(\mathcal{D}_i(u)) = v_i\}.$$

2. Using u, $(\vec{\mathsf{crs}}, \vec{r}) = (\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)$ can be efficiently computed.
3. For $j < i$, set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
4. For $j > i$, if $(\mathsf{crs}_j, r_j) \in \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,no}, \pi_{j,no})$ and if $(\mathsf{crs}_j, r_j) \notin \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
5. Set $(x_i, \pi_i) = \mathcal{L}^*_{\mathsf{crs}_i}(w_i)$.

Assume $(v_i||w_i)$ is either random or the output of a PRG $G_{easy}(v_i)$ that is computed in polynomial time and is secure against NP-circuits of polynomial size $p_1(\lambda)$ (needed to pre-sample u, and check whether $(crs_j, r_j) \in \mathcal{S}$).

If $(v_i||w_i)$ is the output of G_{easy}, then we exactly obtain the Hybrid $H_{i,4}$ distribution, whereas if $(v_i||w_i)$ is random, we exactly obtain the Hybrid $H_{i,3}$ distribution. By the security of the PRG G_{easy} against this complexity class, Hybrid $H_{i,3}$ and Hybrid $H_{i,4}$ are indistinguishable.

Hybrid $H_{i,5}$ In this Hybrid, we go back to sampling $v_i = \mathsf{NMExt}(Z_i(\overrightarrow{\mathsf{crs}})$ from the correct distribution, instead of choosing it uniformly at random. Again, indistinguishability follows from the statistical properties of the $(t-1)$-non-malleable extractor. The formal description of the hybrid follows:

Let $p_i := \Pr_{r_i \sim \mathcal{D}_i[2]}[r_i \in \mathcal{S}]$. With probability p_i do the following:

1. Sample u uniformly at random from the set $\{u : \mathcal{D}_i(u)[2] \in \mathcal{S}\}$. Fix $\mathsf{crs}_i = \mathcal{D}_i(u)[1]$ and fix ρ at random. For $j \in [t] \setminus \{i\}$, fix $\mathsf{crs}_j = D_j(u)[1]$ and $v_j = \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(\mathcal{D}_i(u)))$. Note that the distribution over $\overrightarrow{\mathsf{crs}} := (\mathsf{crs}_1, \ldots, \mathsf{crs}_t)$ is identical to a draw from C_1. Further, $\mathcal{D}_i(u)$ conditioned on $\overrightarrow{\mathsf{crs}}$ is identical to the distribution $Z_i(\overrightarrow{\mathsf{crs}})$.
2. Fix the circuits $C_{\mathsf{crs}_1}, \ldots, C_{\mathsf{crs}_t}$, each of size $\mathsf{poly}(2^{\lambda_\pi})$.
3. For $j > i$, compute $v_j || w_{j,hard} = G_{hard}(v_j)$. Let $w_{j,hard} = w^1_{j,hard} || w^2_{j,hard}$. Set $x_{j,no} = \bar{\mathcal{L}}(w^1_{j,hard}), \pi_{j,no} = C_{\mathsf{crs}_j}(x_{j,no}; w^2_{j,hard})$.
4. For $j \in [t] \setminus \{i\}$, compute $v_j || w_{j,easy} = G_{easy}(v_j)$, Set $(x_{j,yes}, \pi_{j,yes}) = \mathcal{L}^*_{\mathsf{crs}_j}(w_{j,easy})$.
5. Sample $v_i = \mathsf{NMExt}(Z_i(\overrightarrow{\mathsf{crs}}))$, conditioned on $v_j = \mathsf{NMExt}(f_j(Z_i(\overrightarrow{\mathsf{crs}}))), j \in [t] \setminus \{i\}$.
6. Set $(v_i || w_i) = G_{easy}(v_i)$.
7. Pre-sample u uniformly at random from the set

$$\{u : \mathcal{D}_i(u)[2] \in \mathcal{S} \wedge \mathcal{D}_1(u)[1] = \mathsf{crs}_1 \cdots \wedge \mathcal{D}_t(u)[1] = \mathsf{crs}_t \wedge \\ \text{for } j \in [t] \setminus \{i\}, \mathsf{NMExt}(f^i_{j,\overrightarrow{\mathsf{crs}},\rho}(\mathcal{D}_i(u))) = v_j \wedge \mathsf{NMExt}(\mathcal{D}_i(u)) = v_i\}.$$

8. Using u, $(\overrightarrow{\mathsf{crs}}, \overrightarrow{r}) = (\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_t, r_t)$ can be efficiently computed.
9. For $j < i$, set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
10. For $j > i$, if $(\mathsf{crs}_j, r_j) \in \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,no}, \pi_{j,no})$ and if $(\mathsf{crs}_j, r_j) \notin \mathcal{S}$ set $(x_j, \pi_j) = (x_{j,yes}, \pi_{j,yes})$.
11. Set $(x_i, \pi_i) = \mathcal{L}^*_{\mathsf{crs}_i}(w_i)$.

With probability $1 - p_i$, run experiment $H_{i-1,5}$, conditioned on $(\mathsf{crs}_i, r_i) \notin \mathcal{S}$.

Claim.

$$\mathsf{View}^{H_{i,4}}_{\mathsf{aug}} \overset{s}{\approx} \mathsf{View}^{H_{i,5}}_{\mathsf{aug}}$$

Proof. If p_i is negligible, then $H_{i,4}$ and $H_{i,5}$ are both statistically close to the distribution $H_{i-1,5}$, conditioned on $(\mathsf{crs}_i, r_i) \notin \mathcal{S}$ (and hence to each other). On the other hand, if $p_i \geq \frac{1}{2^{\log^2(\lambda)}}$, then the distribution $Z_i(\overrightarrow{\mathsf{crs}})$ has high min-entropy and can be sampled by bounded $\mathsf{poly}(\lambda)$-size Σ_2-circuits (as shown in Section 4.4), and the tampering functions $f^i_{j,\overrightarrow{\mathsf{crs}},\rho}$ can also be computed by $\mathsf{poly}(\lambda)$-size NP-circuits (as shown in Section 4.5). Therefore, due to the relaxed $(t-1)$-non-malleability of NMExt (Def 2), $\mathsf{View}^{H_{i,4}}_{\mathsf{aug}}$ and $\mathsf{View}^{H_{i,5}}_{\mathsf{aug}}$ are $1/\mathsf{poly}$-statistically close.

We claim that Hybrid $H_{i,5}$ is identical to the following hybrid, we simply re-arrange the order of sampling. Specifically, in this Hybrid, we first sample the coins of the reduction and challenger, run the experiment forward. Given all the queries, $(\overrightarrow{\mathsf{crs}}, \overrightarrow{r})$, we run the non-malleable extractor to obtain $v_1, \ldots, v_t$. For $j \leq i$, we run $(v_j, w_j) = G_{easy}(v_j)$ and sample $(x_j, \pi_j) = \mathcal{L}^*_{\mathsf{crs}_j}(w_j)$ from the "yes" distribution. For $j > i$, if the query (crs_j, r_j) is not "frequent", we run $(v_j, w_j) = G_{hard}(v_j)$ and sample $x_j = \overline{\mathcal{L}}(w^1_j)$, $\pi_j = C_{\mathsf{crs}_j}(x_j; w^2_j)$ from the "no" distribution. Otherwise, if the query (crs_j, r_j) is "frequent", we run $(v_j, w_j) = G_{easy}(v_j)$ and sample $(x_j, \pi_j) = \mathcal{L}^*_{\mathsf{crs}_j}(w_j)$ from the "yes" distribution.

Hybrid $H_{i,5}$

1. Sample u uniformly at random and fix u.
2. Using u, $(\overrightarrow{\mathsf{crs}}, \overrightarrow{r}) = (\mathsf{crs}_1, r_1), \ldots, (\mathsf{crs}_2, r_2)$ can be efficiently computed.
3. For $j > i$, fix the circuits C_{crs_j} of size $\mathsf{poly}(2^{\lambda_\pi})$.
4. For $j \in [t]$, set $v_j = \mathsf{NMExt}(\mathsf{crs}_j, r_j)$.
5. For $j \leq i$, set $v_j || w_j = G_{easy}(v_j)$ and set $(x_j, \pi_j) = \mathcal{L}^*_{\mathsf{crs}_j}(w_j)$.

6. For $j > i$, if $(\mathsf{crs}_j, r_j) \in \mathcal{S}$
   (a) Compute $(v_j, w_j) = G_{hard}(v_j)$. Let $w_j = w_j^1 || w_j^2$.
   (b) Set $x_j = \bar{\mathcal{L}}(w_j^1)$
   (c) Set $\pi_j = C_{\mathsf{crs}_j}(x_j; w_j^2)$.
7. For $j > i$, if $(\mathsf{crs}_j, r_j) \notin \mathcal{S}$
   (a) Compute $(v_j, w_j) = G_{easy}(v_j)$. Note that G_{easy} takes polynomial time to compute.
   (b) Set $(x_j, \pi_j) = \mathcal{L}^*_{\mathsf{crs}_j}(w_2)$.

Note that the distribution produced by $H_{-1,5}$ is identical to the one produced by the interaction of R with the real adversary A. Looking at the definition of the distribution in Hybrid $H_{i,0}$, we further claim that for $i \in \{-1, \ldots, t-1\}$, $H_{i,5}$ is identical to $H_{i+1,0}$, with only the order of sampling changed.

Finally, we note that $H_{t,5}$ is a polynomial-time algorithm that produces a distribution over views identical to the one produced by the interaction of R with Sim.

This concludes the proof of Theorem 6.

5 t-Non-Malleable Extractors

5.1 Preliminaries for NMExt Construction

We define functions that will be useful in defining t-non-malleable extractors:

$$\mathrm{Copy}(x, y) = \begin{cases} x & \text{if } x \neq \mathtt{same} \\ y & \text{if } x = \mathtt{same}. \end{cases}$$

For $t \in \mathbb{N}$, we further define

$$\mathrm{Copy}^t(x_1, \ldots, x_t, y) := \mathrm{Copy}(x_1, y) || \cdots || \mathrm{Copy}(x_t, y).$$

Definition 7 (Strong Seeded Extractors). *A function* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k, ϵ)*-strong extractor if for every source* X *over* $\{0,1\}^n$ *with min entropy at least* k *and uniform* Y *over* $\{0,1\}^d$*,* $(Y, \mathsf{Ext}(X, Y)) \approx_\epsilon (Y, U_m)$*, where* U_m *is uniformly distributed over* $\{0,1\}^m$*. Moreover, we require* Ext *to be computable in polynomial time.*

Definition 8 (Strong Two-Source Extractors). *A function* $2\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k, ϵ)*-extractor if for every pair of sources* X, Y *over* $\{0,1\}^n$ *with combined min entropy at least* k*,* $(Y, 2\mathsf{Ext}(X, Y)) \approx_\epsilon (Y, U_m)$*, and* $(X, 2\mathsf{Ext}(X, Y)) \approx_\epsilon (X, U_m)$ *where* U_m *is uniformly distributed over* $\{0,1\}^m$*. Moreover, we require* Ext *to be computable in polynomial time.*

Definition 9 (Seedless t-non-malleable extractors). *Let* $\mathcal{G}$ *be a class of tampering functions* $\{0,1\}^n \to \{0,1\}^n$ *and* $\mathcal{X}$ *be a class of distributions over* $\{0,1\}^n$*. A function* $\mathrm{NMExt} : \{0,1\}^n \to \{0,1\}^m$ *is called an* ϵ*-seedless* t*-non-malleable extractor for source* $\mathcal{X}$ *with respect to tampering class* $\mathcal{G}$ *if for every distribution* $X \in \mathcal{X}$ *and every set of tampering function* $\{g_1, \ldots, g_t\}$*, such that for* $i \in [t]$*,* $g_i \in \mathcal{G}$*, there exists a random variable* $D_{(g_1, \ldots, g_t)}$ *on* $\{0,1\}^m \cup \{\mathtt{same}\}$ *that is independent of* X*, such that*

$$\Delta\Big((\mathrm{NMExt}(X), \mathrm{NMExt}(g_1(X)), \ldots, \mathrm{NMExt}(g_t(X)));$$
$$(\mathcal{U}_m, \mathrm{Copy}^t(D_{(g_1, \ldots, g_t)}, \mathcal{U}_m))\Big) \leq \epsilon.$$

We refer to the parameter ϵ *as the "error" of the seedless* t*-non-malleable extractor.*

Specifically, we say NMExt *is an two-source* (ϵ, t)*-non-malleable extractor for* (n, k)*-sources if it is a* t*-non-malleable extractor for pairs of independent* (n, k)*-sources with respect to split-state tampering, i.e. if for every pair of independent* (n, k)*-sources* X, Y *and split-state tampering functions* $(\tau_L^1, \tau_R^1), \ldots, (\tau_L^t, \tau_R^t)$*, there*

exists a random variable $D_{(\tau_L^1,\tau_R^1),\ldots,(\tau_L^t,\tau_R^t)}$ *supported on* $(\{0,1\}^m \cup \{\mathtt{same}\})^t$ *that is independent of* X, Y*, such that*

$$\Delta((\text{NMExt}(X,Y), \text{NMExt}(\tau_L^1(X),\tau_R^1(Y)),\ldots,\text{NMExt}(\tau_L^t(X),\tau_R^t(Y)));$$
$$(\mathcal{U}_m, \text{Copy}^t(D_{(\tau_L^1,\tau_R^1),\ldots,(\tau_L^t,\tau_R^t)},\mathcal{U}_m))) \leq \epsilon.$$

Definition 10 (Relaxed Seedless t-non-malleable extractor). *Let* $\mathcal{X}$ *be a family of sources on* $\{0,1\}^n$ *and* $\mathcal{F}$ *be a class of tampering functions acting on* $\{0,1\}^n$*. Further assume that all* $f \in \mathcal{F}$ *does not have any fixed points. A function* $\text{NMExt} : \{0,1\}^n \to \{0,1\}^m$ *is defined to be a relaxed* (ϵ, t)*-non-malleable extractor with respect to* $\mathcal{X}$ *and* $\mathcal{F}$ *if the following hold: for any* $X \in \mathcal{X}$ *and any set* $\{f_1,\ldots,f_t\}$*, such that for all* $i \in [t]$*,* $f_i \in \mathcal{F}$*, we have*

$$\Delta\Big((\text{NMExt}(X), \text{NMExt}(f_1(X)),\ldots,\text{NMExt}(f_t(X)));$$
$$(\mathcal{U}_m, \text{NMExt}(f_1(X)),\ldots,\text{NMExt}(f_t(X))\Big) \leq \epsilon.$$

Moreover, if $\mathcal{F}$ *is the class of split-state functions, we say* NMExt *is a relaxed two-source* (ϵ, t)*-non-malleable extractor for independent sources* X, Y *if for every set of* t *pairs of split-state tampering functions* $(\tau_L^1,\tau_R^1),\ldots,(\tau_L^t,\tau_R^t)$*, where for each pair at least one of the functions has no fixed points,*

$$\Delta\Big((\text{NMExt}(X,Y), \text{NMExt}(\tau_L^1(X),\tau_R^1(Y)),\ldots,\text{NMExt}(\tau_L^t(X),\tau_R^t(Y)));$$
$$(\mathcal{U}_m, \text{NMExt}(\tau_L^1(X),\tau_R^1(Y)),\ldots,\text{NMExt}(\tau_L^t(X),\tau_R^t(Y))\Big) \leq \epsilon.$$

Theorem 8 ([12]). *There exists a constant* $\gamma > 0$ *such that for all* $n > 0$ *and* $t \leq n^\gamma$*, there exists an efficient, two-source* (ϵ, t)*-non-malleable extractor for* $(n, n - n^\gamma)$*-sources with error* $\epsilon = 2^{-n^{\Omega(1)}}$ *and output length* $m = n^{\Omega(1)}$*.*

Theorem 9. *Suppose that* $\text{NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *is a two-source* (ε, t)*-non-malleable extractor with error* ε *for* (n,k)*-sources. Then for any* $k' \geq k$*,* NMExt *is a strong, two-source,* (ε', t)*-non-malleable extractor for* (n,k')*-sources with error* $\varepsilon' = 2^{(t+1)m}(\varepsilon + 2^{k+1-k'})$*.*

Proof. Let X_L, X_R be independent (n,k') sources and for $i \in [t]$, let $X_L^i = \tau_L^i(X_L)$, $X_R^i = \tau_R^i(X_R)$, where for each i, f_L^i, f_R^i. are deterministic functions such that at least one of them has no fixed point. Now for any $(z, z_1,\ldots,z_t) \in (\{0,1\}^m)^{t+1}$, define the set of bad x_L's for $\vec{z} = (z, z_1,\ldots,z_t)$ to be

$$B_{\vec{z}} = \Big\{x_L : \big|\Pr[\text{NMExt}(x_L, X_R) = z, \forall j \in [t], \text{NMExt}(\tau_L^j(x_L), X_R^j) = z_j]$$
$$- 2^{-m}\Pr[\forall j \in [t], \text{NMExt}(\tau_L^j(x_L), X_R^j) = z_j]\big| > \epsilon.\Big\}$$

We have the following Claim:

Claim 5.1. For any $\vec{z} = (z, z_1,\ldots,z_t)$, we have $|B_{\vec{z}}| < 2^{k+1}$.

The claim holds since otherwise we can create an (n,k)-source (by placing uniform weight on $B_{\vec{z}}$) which contradicts the original assumption on NMExt.

Now let $B = \bigcup_{\vec{z}} B_{\vec{z}}$ we have that $|B| \leq 2^{(t+1)m}2^{k+1}$. Thus, we now have that

$$
\begin{aligned}
&\big|(\mathrm{NMExt}(X_L, X_R), \mathrm{NMExt}(X_L^1, X_R^1), \ldots, \mathrm{NMExt}(X_L^t, X_R^t), X_L) \\
&\quad - (U_m, \mathrm{NMExt}(X_L^1, X_R^1), \ldots, \mathrm{NMExt}(X_L^t, X_R^t), X_L)\big| \\
&= \sum_{x_L \in \{0,1\}^n} \Pr[X_L = x_L]\big|(\mathrm{NMExt}(x_L, X_R), \mathrm{NMExt}(f_L^1(x_L), X_R^1), \ldots, \mathrm{NMExt}(f_L^t(x_L), X_R^t)) \\
&\quad - (U_m, \mathrm{NMExt}(f_L^1(x_L), X_R^1), \ldots, \mathrm{NMExt}(f_L^t(x_L), X_R^t)))\big| \\
&\leq \Pr[X_L \in B] \cdot 1 + \Pr[X_L \notin B] 2^{(t+1)m}\epsilon \\
&\leq 2^{(t+1)m} 2^{k+1} 2^{-k} + 2^{(t+1)m}\epsilon \\
&= 2^{(t+1)m}(\epsilon + 2^{k+1-k'})
\end{aligned}
$$

Samplable, Recognizable, and Post-selecting Distributions

Definition 11 (Samplable distribution [53, 3].). *We say that a distribution X on n bits is samplable by a class $\mathcal{C}$ of functions $C : \{0,1\}^r \to \{0,1\}^n$ if there exists a function C in the class such that X is distributed as $C(U_r)$.*

Definition 12 (Recognizable distribution [3].). *We say that a distribution X on n bits is recognizable by a class $\mathcal{C}$ of functions $C : \{0,1\}^n \to \{0,1\}$ if there exists a function C in the class such that X is uniform over $\{x : C(x) = 1\}$.*

Definition 13 (Post-selecting distribution [8].). *We say that a distribution X on n bits is samplable by a class $\mathcal{C}$ of functions $C : \{0,1\}^r \to \{0,1\}^n \times \{0,1\}$ via* post-selection *if there exists a function C in the class such that X is distributed as $(C_1(U_r)|C_2(U_r) = 1)$ (where C_1 is the function that outputs the first n-bits of C and C_2 is the function that outputs the last bit.*

We note that any size s samplable or recognizable source can be sampled by a size $s + O(1)$ post-selecting source.

Seedless Extractors for Post-Selecting Sources Ball et al. [8], building on work by Trevisan and Vadhan [53], construct extractors for post-selecting sources from derandomization-type assumptions.

Theorem 10 ([8]). *If E requires exponential size nondeterministic circuits, then there exists a constant $\alpha > 0$ such that for every constant $c > 1$ and sufficiently large n, and there is a $((1-\alpha)n, n^{-c})$-extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^{\alpha n}$ for $\mathsf{SIZE}[n^c]$-post-selecting sources. Moreover, Ext is computable in time $\mathsf{poly}(n^c)$.*

Moreover, because this result relativizes, one the following more generally holds with respect to any oracle O: If E requires exponential size nondeterministic O-oracles circuits, then there exists a constant $\alpha > 0$ such that for every constant $c > 1$ and sufficiently large n, and there is a $((1-\alpha)n, n^{-c})$-extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^{\alpha n}$ for $\mathsf{SIZE}^O[n^c]$-post-selecting sources. Moreover, Ext is computable in time $\mathsf{poly}(n^c)$.

Interactive Proofs We require some facts about interactive proofs. We refer a curious reader unfamiliar with this topic to the excellent presentation in Arora and Barak's [4, Chapter 8].

Proposition 1 (Implicit in Lemma 3.8.1 in [55]). *Let X_0, X_1 be random variables such that $\Delta(X_0; X_1) = \epsilon$. Consider the following game:*

- *Arthur samples a coin $b \leftarrow \mathcal{U}$ and gives Merlin $x \leftarrow X_b$.*
- *Merlin responds with b'. If $b' = b$, Merlin wins. Otherwise, Merlin loses.*

Merlin wins with probability $\frac{1+\epsilon}{2}$ by outputting b' such that $\Pr[X_{b'} = x] \geq \Pr[X_{1-b'} = x]$. Moreover, this strategy is optimal.

Promise problems generalize the concept of languages that give a better handle on semantic complexity classes. A promise problem, Π, consists of a set of Yes instances, Π_Y, and a disjoint set of No instances, Π_N. A machine is considered to decide Π if on input x *promised* to be in $\Pi_Y \cup \Pi_N$ it accepts x if and only if $x \in \Pi_Y$. In other words, the machine should accept Π_Y and reject Π_N, but can behave arbitrarily elsewhere.

To reduce assumptions, we observe that the private coin to public coin transformation of Goldwasser and Sipser [26], and a recent improvement by Goldreich and Leshkowitz [23], applies to not just simple private coin protocols, but protocols where Arthur can sample from a recognizable source that may depend on the input, provided the recognizable sources have bounded entropy. Put differently, as long as there is a sufficiently large multiplicative gap between the minimum number of accepting coins for Yes instances and the maximum number of accepting coins for No instances, then the promise problem admits a public coin protocol. What is potentially somewhat surprising is that this holds even when the number of accepting coin sequences is negligibly small relative to the total number potential coin sequences, and thus Arthur could not hope to sample such coins on his own. Nonetheless, with the assistance of Merlin, Arthur can efficiently sample such coins (without negatively impacting soundness).

Theorem 11 ([23], Theorem 2). *There is a constant B such that the following holds.*

If $\Pi = (\Pi_Y, \Pi_N)$ is a promise problem such that Π admits an r-round interactive proof system where

$$\frac{\min_{x \in \Pi_Y} |\{r : \langle P(x), V(x;r)\rangle = \text{“}\mathtt{acc}''\}|}{\max_{P^*, x \in \Pi_N} |\{\{r : \langle P^*(x), V(x;r)\rangle = \text{“}\mathtt{acc}''\}|} \geq B^r,$$

then there exists an $(r+2)$-round public-coin proof system for Π with soundness/completeness error 1/3.

From this theorem, we can deduce that any promise problem Π that admits a constant round interactive proof as above, can be recognized by polynomial size nondeterministic circuits. This follows by applying two additional classical transformations. First, gap amplification to transform an inverse polynomial gap to an arbitrarily large one. Next, apply the transformation of [5, 6] who show that any constant round public coin interactive proof can be transformed into an *AM* protocol. Finally, applying Adleman's trick [1] to the resulting AM protocol yields a *non-uniform*, non-deterministic circuit at most polynomially larger than Arthur's original complexity.

Theorem 12 ([26, 23, 5, 6, 1]). *Let c be any constant.*

If $\Pi = (\Pi_Y, \Pi_N)$ is a promise problem such that Π admits an c-round interactive proof system where

$$\frac{\min_{x \in \Pi_Y} |\{r : \langle P(x), V(x;r)\rangle = \text{“}\mathtt{acc}''\}|}{\max_{P^*, x \in \Pi_N} |\{\{r : \langle P^*(x), V(x;r)\rangle = \text{“}\mathtt{acc}''\}|} \geq B^r,$$

then Π is recognized by a nondeterministic circuit of size $n^{O(1)}$.

Useful Propositions We also use the following simple combinatorial propositions.

Proposition 2. *Let X be a random variable and f a function. Define $Y = f(X)$. For any ϵ and any random variable Y',*

$$\Delta(X; (X|f(X) = Y')) = \Delta(Y; Y').$$

Proof. Let $X' \equiv (X|f(X) = Y')$.

$$\begin{aligned}
\Delta(X;X') = \Delta(XY;X'Y') &= \frac{1}{2}\sum_{x,y} |\Pr[Y=y]\Pr[X=x|Y=y] - \Pr[Y'=y]\Pr[X'=x|Y'=y]| \\
&= \frac{1}{2}\sum_{x,y} |\Pr[Y=y]\Pr[X=x|f(X)=y] - \Pr[Y'=y]\Pr[X'=x|f(X')=y]| \\
&= \frac{1}{2}\sum_{x,y} |\Pr[Y=y]\Pr[X=x|f(X)=y] - \Pr[Y'=y]\Pr[X'=x|f(X)=y]| \\
&= \frac{1}{2}\sum_{x,y} |\Pr[Y=y] - \Pr[Y'=y]|\Pr[X=x|f(X)=y] \\
&= \frac{1}{2}\sum_{y} |\Pr[Y=y] - \Pr[Y'=y]|\sum_{x}\Pr[X=x|f(X)=y] \\
&= \frac{1}{2}\sum_{y} |\Pr[Y=y] - \Pr[Y'=y]| \\
&= \Delta(Y;Y')
\end{aligned}$$

Proposition 3. *Let $c > 1$. Let $(XY), (XZ)$ be two joint random variables supported on any space $\Sigma_X \times \Sigma$ such that $\Delta(X,Y;X,Z) \leq \epsilon$, then there exists an event $S \subseteq \Sigma_X$ such that*

1. $\Pr[X \in S] \geq 1 - 1/c$
2. $\forall x \in S, \Delta(XY|X=x; XZ|X=x) \leq c\epsilon$, *where $(XY|X=x)$ denotes the random variable XY conditioned on $X=x$ and, similarly, $(XZ|X=x)$ denotes the random variable XZ conditioned on $X=x$.*

Proof. Let A the variable distributed according to the procedure where $x \leftarrow X$ and then $\Delta(XY|X=x; XZ|X=x)$ is output. By definition, $\mathbb{E}[A] \leq \epsilon$.[9] Thus, by Markov's inequality we have

$$\Pr[A \geq c\epsilon] \leq \frac{\mathbb{E}[A]}{c\epsilon} \leq \frac{1}{c}$$

It follows that there exists a set S with the desired properties. In particular, S is the set of x such that A conditioned on $X = x$ is *not* greater than $c\epsilon$.

Proposition 4. *Let $\beta \in (0,1)$ Let $(XY), (XZ)$ be two joint random variables supported on any space $\Sigma_X \times \Sigma$ such that $\Delta(X,Y;X,Z) > \epsilon$, then exists an event $S \subseteq \Sigma_X$ such that*

1. $\Pr[X \in S] \geq \epsilon - \beta$
2. $\forall x \in S, \Delta(XY|X=x; XZ|X=x) \geq \beta$, *where $(XY|X=x)$ denotes the random variable XY conditioned on $X=x$ and, similarly, $(XZ|X=x)$ denotes the random variable XZ conditioned on $X=x$.*

Proof. Let A the variable distributed according to the procedure where $x \leftarrow X$ and then $\Delta(XY|X=x; XZ|X=x)$ is output. Let $A' = 1 - A$. By definition, $\mathbb{E}[A'] < 1 - \epsilon$. Thus, by Markov's inequality we have,

$$\Pr[A' \geq 1 - \beta] < \frac{1-\epsilon}{1-\beta}.$$

So, it follows that $\Pr[A \leq \beta] < \frac{1-\epsilon}{1-\beta}$. Thus

$$\Pr[A > \beta] \geq 1 - \frac{1-\epsilon}{1-\beta} = \frac{\epsilon - \beta}{1-\beta} \geq \epsilon - \beta.$$

It follows that there exists a set S with the desired properties. In particular, S is the set of x such that A conditioned on $X = x$ is greater than β.

[9] $\mathbb{E}[A] = \sum_{x \in \Sigma_X} \Pr[X=x]\Delta(XY|X=x; XZ|X=x) = \Delta(XY;XZ)$.

5.2 Relaxed t-Non-Malleable Extractors for (Classical) Postselecting Sources

In this section, we begin to construct our seedless non-malleable extractors for samplable and recognizable sources resistant to tampering by polynomial-size circuits. In particular, we first construct *relaxed* non-malleable extractors for the special case that tampering functions contain no fixed points. In later sections, we will so how to generically remove this restriction by generalizing a connection observed by Cheraghchi and Guruswami [14] to the case of post-selecting circuits.

We follow the construction laid out by Ball et al. [7]:starting with a source x, (a) extract a short seed, $\mathsf{Ext}_{\text{samp}}(x) = \sigma$, with an extractor for post-selecting sources ($\mathsf{Ext}_{\text{samp}}$, Theorem 10), (b) expand σ with a seed-extending PRG $G(\sigma) = (\sigma, y)$ for nondeterministic circuits (G, Theorem 3), (c) apply a (strong) non-malleable two-source extractor to get the output $z = \text{2NMExt}((\sigma, y), x)$ (2NMExt, Theorem 8 and Theorem 9). The only difference between our construction and the prior one is that the we replace the strong non-malleable two-source extractor (c) with strong t-non-malleable two-source extractor.

Our analysis follows the same high-level template as that of Ball et al. [7] as well: reduce a non-malleability attack to nondeterministic PRG distinguisher by constructing a constant round interactive proof "distinguisher" for the PRG using the non-malleability violation, and then "compile down" the IP into a (nondeterministic) circuit. However, while the initial IP is very similar to that of Ball et al., we provide a different fine-grained analysis using a new insight into the surprising power of private-to-public coin emulations [26, 23] (see Theorem 12). Our initial distinguisher is not yield a sufficient large completeness/soundness gap to immediately apply this theorem, however a simple amplification step suffices to yield the desired parameters.

Figure 5.1: Non-Malleable Extractor for Postselecting Sources

Let $k(n), s(n), s'(n), \gamma$ be as in Lemma 1. Let $\mathsf{Ext}_{\text{samp}}$ be an extractor with error $\gamma(n)$ for n-bit $s(n)$-postselecting (classical) sources, computable in time $\mathsf{poly}(s(n))$. Let 2NMExt be a strong relaxed two-source t-non-malleable extractor with error $\delta(n)$ for independent sources of length n where the left has min-entropy at least $n-\ell(n)$ and the right has min-entropy at least $k(n)-(t+1)\ell(n)-3\log(s(n))-11$, computable in time $\mathsf{poly}(s)$. Let G be a seed-extending PRG for nondeterministic circuits of size $s'(n)$.

$$\mathsf{NMExt}_{\text{samp}} : x \mapsto \text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(x)), x)$$

Lemma 1. *For any polynomial $s(n)$ and function $k(n)$ such that $0 \le k(n) \le n$, there exists polynomial $s'(n) = \Omega(s(n))$ such that the following is true.*

If

- $\mathsf{G} : \{0,1\}^{\ell(n)} \to \{0,1\}^n$ *is a seed-extending PRG for nondeterministic* NP*-circuits of size $s'(n)$ with seed length $\ell(n)$.*
- $\mathsf{Ext}_{samp} : \{0,1\}^n \to \{0,1\}^\ell(n)$ *is a γ-extractor for (n,k) sources samplable by $s(n)$-size circuits computable in time* $\mathsf{poly}(s(n))$*, where $\gamma \le 1/6s(n)$.*
- $\text{2NMExt} : \{0,1\}^{2n} \to \{0,1\}^m$ *is a strong relaxed two-source t-non-malleable extractor with error $\delta(n) < 1/1000(s(n))^2$ for two independent n-bit sources where the left source has min-entropy at least $n-\ell(n)$ and the right has min-entropy at least $k(n)-(t+1)\ell(n)-3\log(s(n))-11$). Moreover,* 2NMExt *should be computable in time* $\mathsf{poly}(s(n))$*.*

then the construction, $\text{NMExt} : \{0,1\}^n \to \{0,1\}^m$*, in Figure 5.1 is a relaxed seedless non-malleable extractor for n-bit sources with $k(n)$-min entropy samplable by size $s(n)$ circuits with respect to* $\mathsf{SIZE}[s(n)]$*-tampering and error $1/s(n)$.*

Proof. Let $\epsilon = 1/s(n)$.

Suppose for the sake of contradiction that there exists a $s(n)$-samplable (n,k)-source X and t tampering functions, $\tau_1, \ldots, \tau_t : x \mapsto \tilde{x}$ in $\mathsf{SIZE}[s(n)]$ with no fixed points, that breaks the non-malleability guarantee.

Now, our assumption on τ can be restated as

$$\Delta(\text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(X)), X), \text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(\tau(X))), \tau(X));$$
$$\mathcal{U}_m, \text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(\tau(X))), \tau(X)) \ge \epsilon.$$

We will use this assumption to distinguish the seed-extending PRG, G, from the uniform distribution via an interactive proof. In more detail, recall that the guarantee of $\mathsf{G} : \{0,1\}^\ell \to \{0,1\}^n$ says that for any non-deterministic NP circuit, C, of size $s'(n)$,

$$\Delta(C(\mathsf{G}(\mathcal{U}_\ell)); C(\mathcal{U}_n)) < 1/s'(n).$$

We show that there exists a circuit C of size at most $s'(n)$ that does not obey this inequality. We do this by following the approach of [7, 3] and constructing a private coin, constant round interactive proof protocol (see Figure 5.2) where Arthur is an NP-circuit of size at most $\mathsf{poly}(s(n))$ for a promise problem, $\Pi = (\Pi_Y, \Pi_N)$ where Π_Y is dense under G and Π_N is dense under the uniform distribution. By demonstrating a sufficiently large completeness/soundness gap we can apply Theorem 12, to yield a nondeterministic NP-circuit of size $s'(n) = \mathsf{poly}(s(n))$ that decides the same problem, and hence breaks the PRG.

Looking ahead, the *malleability* the two-source extractor when provided pseudorandom inputs (apropos our assumption) will enable us to prove the protocol is complete, i.e. Arthur accepts pseudorandom inputs with high probability. Soundness, i.e. Arthur rejects random inputs with high probability, will ultimately follow from security of the 2-source non-malleable extractor.

To do this, Arthur will run the non-malleability experiment himself, using Merlin to evaluate the PRG. In order to render the experiment consistent with the PRG seed in the PRG security game, s, Arthur will attempt to sample X conditioned on the samplable-source extractor outputting s. If this fails, Arthur simply fails (and rejects). Armed with a consistent seed, Arthur attempts to complete the real and ideal tampering experiments. Because G is to expensive for him to compute, he asks Merlin to evaluate G on the tampered seeds for him. Because the collective information sent to Merlin remains short (and hence has limited dependence on Arthur's secret X, any misbehaving Merlin can ultimately be cast as a split-state tampering to bound soundness.

We then conclude by using a standard private-coin technique [24], to distinguish between the real and ideal relaxed non-malleable extractor experiments, again with Merlin's assistance.

Figure 5.2: Interactive Proof for distinguishing G from uniformly random bits

Let $\mathsf{Ext}_{\mathrm{samp}}$ be an extractor with error $\gamma(n)$ for n-bit $s(n)$-postselecting sources, computable in time $s'(n) = \mathsf{poly}(s(n))$. Let 2NMExt be a strong two-source t-non-malleable extractor with error $\delta(n)$ for independent sources of length n where the left has min-entropy at least $n - \ell(n)$ and the right has min-entropy at least $k - (t+1)\ell(n) - 2\log(s(n)) - 10$, computable in time $s'(n) = \mathsf{poly}(s(n))$. Let G be a seed-extending PRG for ???? circuits of size $O(s'(n))$.
Recall that X is the $s(n)$-postselecting source, corresponding to a circuit C and $\tau_1, \ldots, \tau_t$ the tampering attack from our assumption.
Our protocol aims to accept strings from $\mathsf{G}(\mathcal{U}_\ell)$ when Merlin plays according to below (completeness) and reject strings from $\mathcal{U}_n$ regardless of the strategy Merlin utilizes (soundness).

On input (σ, y)**,**

Arthur Attempt to sample $x \leftarrow X$ by sampling $r \xleftarrow{u} \mathcal{U}$ and evaluating $C(r) = (x, \phi)$ where ϕ is the post-select bit. If $\mathsf{Ext}_{\mathrm{samp}}(x) \neq \sigma$ or $b = 0$, output $\perp$ and reject.
Otherwise, set $\tilde{x}_1 = \tau_1(x), \ldots, \tilde{x}_t = \tau_t(x)$ and send Merlin $\tilde{\sigma}_1 = \mathsf{Ext}_{\mathrm{samp}}(\tilde{x}_1), \ldots, \tilde{\sigma}_t = \mathsf{Ext}_{\mathrm{samp}}(\tilde{x}_t)$.

Merlin If $(\sigma, y) = \mathsf{G}(\sigma)$, respond $\tilde{y}$ such that $(\tilde{\sigma}_1, \tilde{y}_1) = \mathsf{G}(\tilde{\sigma}_1), \ldots, ((\tilde{\sigma}_t, \tilde{y}_t) = \mathsf{G}(\tilde{\sigma}_t)$. Otherwise, respond using any fixed $\tilde{y}_1, \ldots, \tilde{y}_t$.

Arthur Sample a random coin $b \leftarrow \mathcal{U}$ and set $\tilde{z}_1 = \mathrm{2NMExt}((\tilde{\sigma}_1, \tilde{y}_1), \tilde{x}_1), \ldots, \tilde{z}_t = \mathrm{2NMExt}((\tilde{\sigma}_t, \tilde{y}_t), \tilde{x}_t)$.
- *If* $b = 0$*:* Sample $z \leftarrow \mathcal{U}_m$ and send $z, \tilde{z}_1, \ldots, \tilde{z}_t$.
- *Else if* $b = 1$*:* Set $z = \mathrm{2NMExt}((\sigma, y), x)$ and send $z, \tilde{z}_1, \ldots, \tilde{z}_t$.

Merlin (Guess Arthur's bit.) If

$$\Pr_{\mathcal{U}_m, X}\left[\begin{array}{c|c} \mathcal{U}_m = z, & \mathsf{Ext}_{\mathrm{samp}}(X) = \sigma, \\ \mathrm{2NMExt}((\tilde{\sigma}_1, \tilde{y}_1) = \tilde{z}_1, & \mathsf{Ext}_{\mathrm{samp}}(\tau_1(X)) = \tilde{\sigma}_1, \\ \vdots & \vdots \\ \mathrm{2NMExt}((\tilde{\sigma}_t, \tilde{y}_t), \tau_t(X)) = \tilde{z}_t & \mathsf{Ext}_{\mathrm{samp}}(\tau_t(X)) = \tilde{\sigma}_t \end{array}\right]$$

is upper bound by

$$\Pr_{\mathcal{U}_m, X}\left[\begin{array}{c|c} \mathrm{2NMExt}((\sigma, y), X) = z, & \mathsf{Ext}_{\mathrm{samp}}(X) = \sigma, \\ \mathrm{2NMExt}((\tilde{\sigma}_1, \tilde{y}_1) = \tilde{z}_1, & \mathsf{Ext}_{\mathrm{samp}}(\tau_1(X)) = \tilde{\sigma}_1, \\ \vdots & \vdots \\ \mathrm{2NMExt}((\tilde{\sigma}_t, \tilde{y}_t), \tau_t(X)) = \tilde{z}_t & \mathsf{Ext}_{\mathrm{samp}}(\tau_t(X)) = \tilde{\sigma}_t \end{array}\right]$$

set $b' = 1$. Otherwise, set $b' = 0$. Respond b'.

Arthur Accept if $b = b'$, and reject otherwise.

To avoid redundant, lengthy notation, we introduce the following joint tampering, jointly tampered sources, and various forms of joint extraction and pseudorandom expansion:

$$\begin{aligned}
\overline{\tau}(X) &:= (\overline{\tau}_1(X), \ldots \overline{\tau}_t) = \overline{X} \\
\overline{\mathsf{Ext}_{\mathrm{samp}}}(X_1, \ldots, X_t) &:= (\mathsf{Ext}_{\mathrm{samp}}(X_1), \ldots, \mathsf{Ext}_{\mathrm{samp}}(X_t)) \\
\overline{\mathsf{NMExt}_{\mathrm{samp}}}(X_1, \ldots, X_t) &:= (\mathsf{NMExt}_{\mathrm{samp}}(X_1), \ldots, \mathsf{NMExt}_{\mathrm{samp}}(X_t)) \\
\overline{\mathrm{2NMExt}}(L_1, R_1, \ldots, L_t, R_t) &:= (\mathrm{2NMExt}(L_1, R_1), \ldots, \mathrm{2NMExt}(L_t, R_t)) \\
\overline{G}(\sigma_1, \ldots, \sigma_t) &:= (G(\sigma_1), \ldots, G(\sigma_t))
\end{aligned}$$

Before we begin we will need to eliminate the set of "bad" seeds, these are values of $\sigma \in \Sigma$ on which $\mathsf{Ext}_{\mathrm{samp}}(X)$ deviates significantly from uniform:

$$\begin{aligned}
S_\alpha &:= \{\sigma : \Pr[\mathsf{Ext}_{\mathrm{samp}}(X) = \sigma] \notin (1 \pm \delta)2^{-\ell}\} = S_\alpha^+ \cup S_\alpha^- \\
S_\alpha^+ &:= \{\sigma : \Pr[\mathsf{Ext}_{\mathrm{samp}}(X) = \sigma] \geq (1+\delta)2^{-\ell}\} \\
S_\alpha^- &:= \{\sigma : \Pr[\mathsf{Ext}_{\mathrm{samp}}(X) = \sigma] \leq (1-\delta)2^{-\ell}\}
\end{aligned}$$

We claim that for any $\alpha > 0$, $\Pr[\Sigma \in S_\delta] \leq \frac{2\gamma}{\alpha}$. To see this,

$$\begin{aligned}
\gamma &\geq \frac{1}{2}\sum_\sigma |\Pr[\mathsf{Ext}_{\mathrm{samp}}(X) = \sigma] - 2^{-\ell}| \\
&\geq \frac{1}{2}\sum_{\sigma \in S_\alpha^+} |(1+\alpha)2^{-\ell} - 2^{-\ell}| + \frac{1}{2}\sum_{\sigma \in S_\alpha^-} |(1-\alpha)2^{-\ell} - 2^{-\ell}| \\
&= \frac{1}{2}\sum_{\sigma \in S} \alpha \cdot 2^{-\ell} \\
&= \frac{\alpha}{2}\sum_{\sigma \in S} 2^{-\ell} = \frac{\alpha}{2}\Pr[\Sigma \in S_\delta]
\end{aligned}$$

Define p to be the probability postselection succeeds with a uniformly random input.

$$p := \Pr_r[C(r) = (x, 1)]$$

Claim 5.2. (Completeness) For any $\beta \in (0,1), \alpha > 0$, there exists a set Π_Y^β such that

1. Π_Y is noticeably dense in G: $\Pr_{\sigma \xleftarrow{u} \{0,1\}^\ell}[\mathsf{G}(\sigma) \in \Pi_Y^\beta] \geq \epsilon - \gamma - 2\gamma/\alpha - \beta$.
2. Arthur accepts inputs in Π_Y^β with probability $> \frac{(1-\alpha)(1+\beta)p}{2^{\ell+1}}$ when playing with (honest) Merlin (as prescribed in Figure 5.2).

Proof. We begin by considering how the protocol behaves on *random* inputs distributed according to $(\Sigma, \mathsf{G}(\Sigma))$ where Σ is uniform ($\Sigma \equiv \mathcal{U}_\ell$), *conditioned on* Arthur *not* outputting $\perp$ (i.e. conditioned on sampling x from $X|\mathsf{Ext}_{\mathrm{samp}}(X) = \Sigma$ in the first step).

In particular, by the guarantee of $\mathsf{Ext}_{\mathrm{samp}}$, we have that $\mathsf{Ext}_{\mathrm{samp}}(X) \approx_\gamma \mathcal{U}_\ell \equiv \Sigma$. It follows from Proposition 2 that

$$(\Sigma, X|\mathsf{Ext}_{\mathrm{samp}}(X) = \Sigma) \approx_\gamma (\mathsf{Ext}_{\mathrm{samp}}(X), X).$$

Now, we will additionally consider conditioning on Σ not being "bad" for $\mathsf{Ext}_{\mathrm{samp}}$. Namely, we let Σ' denote $(\Sigma|\Sigma \notin S_\alpha)$. By $\Pr[\Sigma \in S_\alpha] \leq 2\gamma/\alpha$, have that $\Sigma \approx_{2\gamma/\alpha} \Sigma'$ and moreover,

$$(\Sigma', X|\mathsf{Ext}_{\mathrm{samp}}(X) = \Sigma') \approx_{2\gamma/\alpha+\gamma} (\mathsf{Ext}_{\mathrm{samp}}(X), X)$$

Now if take X' to denote $(X|\mathsf{Ext}_{\mathrm{samp}}(X) = \Sigma')$ it follows from postprocessing that

$$\begin{aligned}
&\big(\mathsf{Ext}_{\mathrm{samp}}(X), G(\mathsf{Ext}_{\mathrm{samp}}(X)), \overline{\mathsf{Ext}_{\mathrm{samp}}}(\overline{\tau}(X)), \mathsf{NMExt}_{\mathrm{samp}}(X), \overline{\mathsf{NMExt}_{\mathrm{samp}}}(\overline{\tau}(X))\big) \\
&\approx_\gamma \big(\Sigma', G(\Sigma'), \overline{\mathsf{Ext}_{\mathrm{samp}}}(\Sigma'), \mathsf{NMExt}_{\mathrm{samp}}(X'), \overline{\mathsf{NMExt}_{\mathrm{samp}}}(\overline{\tau}(X'))\big)
\end{aligned}$$

Thus (continuing to condition on Arthur not outputting $\perp$), observe that if Arthur chooses $b = 1$, it follows from the above that Arthur's last message

$$\big(\mathsf{NMExt}_{\mathrm{samp}}(X'), \overline{\mathsf{NMExt}_{\mathrm{samp}}}(\overline{\tau}(X'))\big)$$

is $2\gamma/\alpha + \gamma$-close to

$$\left(\mathsf{NMExt}_{\text{samp}}(X), \overline{\mathsf{NMExt}_{\text{samp}}}(\overline{\tau}(X))\right).$$

On the other hand (still conditioning on not $\perp$), if $b = 0$, Arthur's last message

$$\left(\mathcal{U}_m, \overline{\mathsf{NMExt}_{\text{samp}}}(\overline{\tau}(X'))\right)$$

is $2\gamma/\alpha + \gamma$-close to

$$\left(\mathcal{U}_m, \overline{\mathsf{NMExt}_{\text{samp}}}(\overline{\tau}(X))\right).$$

By our assumption, these latter two joint distributions are ϵ-far from each other.

By Proposition 4 and triangle inequality, this implies there exists a set Π_Y^β such that for any $(\sigma, y) \in \Pi_Y^\beta$ the distributions of Merlin's view in the case that $b = 1$ is β-far from his view when $b = 0$ (conditioned on Arthur not outputting $\perp$ in both cases), and moreover $\Pr[(\Sigma, \mathsf{G}(\Sigma)) \in \Pi_Y^\beta] \geq \epsilon - \gamma - 2\gamma/\alpha - \beta$.

So by Proposition 5.1, for any $(\sigma, \mathsf{G}(\sigma)) \in \Pi_Y^\beta$, conditioned on Arthur not outputting $\perp$ initially, Merlin guesses correctly with probability $\geq \frac{1+\beta}{2}$.

Finally, dealing with Arthur's sampling, we see that Arthur fails to output $\perp$, i.e samples X such that $\mathsf{Ext}_{\text{samp}}(X) = \Sigma'$, with $\Pr[\mathsf{Ext}_{\text{samp}}(X) = \Sigma] \geq (1-\alpha)2^{-\ell}$. Namely, for any $\sigma \notin S_\alpha$,

$$\begin{aligned}
\Pr[\text{sampling succeeds}] &= \Pr_{(x,\phi))\leftarrow C(\mathcal{U})}[x = 1 \wedge \mathsf{Ext}_{\text{samp}}(x) = \sigma] \\
&= \Pr_{x,\phi}[\mathsf{Ext}_{\text{samp}}(x) = \sigma | \phi = 1] \Pr_{\phi}[\phi = 1] \\
&\geq \frac{1-\alpha}{2^m} \cdot p
\end{aligned}$$

$$\begin{aligned}
\forall (s, y) \in \Pi_Y^\beta, \quad \Pr[\text{Arthur accepts}(s, y)] &\geq \Pr[\text{sampling succeeds}] \frac{1+\beta}{2} \\
&\geq \frac{(1-\alpha)(1+\beta)}{2 \cdot 2^\ell} p
\end{aligned}$$

Claim 5.3. For any $c > 1$ and $\zeta(n) \in (0, 1)$ such that $k'(n) \leq k(n) - 2\ell(n) - \log(1/\zeta(n))$ (where $k'(n)$ is the min-entropy requirement of the right source for 2NMExt), there exists a set Π_N^c such that

1. Π_N^c is large: $\Pr_{(\sigma,y)\xleftarrow{u}\{0,1\}^n}[(\sigma, y) \in \Pi_N^c] \geq 1 - 1/c$
2. Arthur accepts inputs in Π_N^c with probability $\leq \frac{p(1+\alpha)(1+c(\delta+\zeta+2\gamma/\alpha)}{2^{\ell+1}}$ when playing with any (cheating) Merlin (as prescribed in Figure 5.2).

Proof. As in Claim 5.2, we will analyze the view of Merlin (up to guessing) on a random input and deduce that there exists a large Π_N which Arthur fails to accept with significant probability. The important difference is that, here, Merlin can behave arbitrarily.

Observe that, that if we condition on success in Arthur's initial sampling, Arthur accepts if and only if Merlin guesses his bit, b, correctly ($b' = b$). It follows by Proposition 5.1 that there is an optimal (for any specific input, not just with respect to uniform inputs) Merlin strategy, M^*, that chooses messages to maximize the distance between his view when Arthur chooses $b = 0$ versus his view when $b = 1$. By the optimality of such a strategy, it suffices to consider just this M^*.

In wat follows, as in the proof of the previous claim, we will condition on Arthur's sampling succeeding (not outputting $\perp$) until the very end.

So, suppose the protocol in Figure 5.2 is given uniformly random inputs $(\sigma, y) \leftarrow \mathcal{U}_n$. Fix an optimal strategy, M^*. In particular, let $G^* : (\sigma, y, \overline{\sigma}) \mapsto \overline{y}$ be the function that given the transcript thus far, outputs Merlin's first message.

Now, note that if we condition on $\sigma, \overline{\sigma}$, then $G^*(\sigma, y, \overline{\sigma}) = \overline{y}$ is independent of x, the string sampled by Arthur initially. And similarly, after conditioning on σ and $\overline{\sigma}$, $\overline{x} = \overline{\tau}(x)$ is independent of (σ, y). In other words, we can sample $(\sigma, y, x, \overline{\sigma}, \overline{y}, \overline{x})$ identically as follows:

1. Sample σ uniformly at random and $\overline{\sigma} \leftarrow \overline{\mathsf{Ext}_{\mathrm{samp}}}(\overline{\tau}(X_\sigma))$, where $X_\sigma \equiv X|\mathsf{Ext}_{\mathrm{samp}}(X) = \sigma$. (This is identically distributed to Figure 5.2.) Let $\Sigma, \overline{\Sigma}$ denote these random variables.
2. Sample y uniformly at random, and sample x from $X_\sigma|\overline{\mathsf{Ext}_{\mathrm{samp}}}(\overline{\tau}(X_\sigma)) = \overline{\sigma}$. Note that conditioned on σ and $\overline{\sigma}$, x is independent of y. Let Y, $X_{\sigma,\overline{\sigma}}$ denote the random variables corresponding to how y and x are sampled here, respectively.
3. Apply the tamperings, for $i = 1, \ldots, t$:
   - $\tau_{L,i}^{\sigma,\overline{\sigma}} : (\sigma, y) \mapsto \overline{\sigma}_i, \overline{y}_i$ where $\overline{y} = G^*(\sigma, y, \overline{\sigma})$
   - $\tau_{R,i}^{\sigma,\overline{\sigma}} : x \mapsto \overline{x}_i = \tau_i(x)$

   Thus, conditioned on σ and $\overline{\sigma}$,

$$\overline{\tau}_{L,R}^{\sigma,\overline{\sigma}} = (\tau_{L,1}^{\sigma,\overline{\sigma}}, \tau_{R,1}^{\sigma,\overline{\sigma}}), \ldots, (\tau_{L,t}^{\sigma,\overline{\sigma}}, \tau_{R,t}^{\sigma,\overline{\sigma}})$$

   is a split-state tampering. Moreover, because $\overline{\tau}$ has no fixed points, neither does any $\tau_{R,i}^{\sigma,\overline{\sigma}}$.

Thus, for any (valid) fixed choice of $\sigma, \overline{\sigma}$, $Y, X_{\sigma,\overline{\sigma}}$ are independent, and $\overline{\tau}_{L,R}^{\sigma,\overline{\sigma}}$ are t split-state tampering functions with no fixed points. Clearly, Y is always uniformly distributed, so Y has min-entropy $n - \ell$ for any fixed choice of σ. Intuitively, $X_{\sigma,\overline{\sigma}}$ should have not lost much min-entropy on average relative to X because $\sigma, \overline{\sigma}$ are not too long. We next formalize this intuition.

For any $\zeta \in (0, 1)$, let $T_{\zeta,\alpha}$ denote the set of $(\sigma, \overline{\sigma})$ that occur with probability at least $\zeta \cdot 2^{-(t+1)\ell}$ and $\sigma \notin S_\alpha$. Let $T'_\zeta \supseteq T_{\alpha,\zeta}$ denote the set of $(\sigma, \overline{\sigma})$ that occur with probability at least $\zeta \cdot 2^{-(t+1)\ell}$ Note that $\Pr[(\Sigma, \overline{\Sigma}) \notin T'_\zeta] \leq \zeta$, because

$$\Pr[(\Sigma, \overline{\Sigma}) \notin T'_{\alpha,\zeta}] \leq \sum_{(\sigma,\overline{\sigma})} \zeta \cdot 2^{-(t+1)\ell} = 2^{(t+1)\ell} \zeta 2^{-(t+1)\ell} = \zeta.$$

So, it follows that $\Pr[(\Sigma, \overline{\Sigma} \in T_{\alpha,\zeta}] > 1 - \zeta - 2\gamma/\alpha$.

Now, for any $(\sigma, \overline{\sigma}) \in T_{\alpha,\zeta}$ and any $x \in \{0,1\}^n$, we have

$$\begin{aligned}
\Pr[X_{\sigma,\overline{\sigma}} = x] &= \Pr[X = x | (\Sigma, \overline{\Sigma}) = (\sigma, \overline{\sigma})] \\
&\leq \frac{\Pr[X = x \wedge (\Sigma, \overline{\Sigma}) = (\sigma, \overline{\sigma})]}{\Pr[(\Sigma, \overline{\Sigma}) = (\sigma, \overline{\sigma})]} \\
&\leq \frac{\Pr[X = x]}{\Pr[(\Sigma, \overline{\Sigma}) = (\sigma, \overline{\sigma})]} \\
&\leq \frac{2^{-k}}{\zeta 2^{-(t+1)\ell}}
\end{aligned}$$

Thus, for any $(\sigma, \overline{\sigma}) \in T_{\alpha,\zeta}$ (which happens with probability at least $1 - \zeta$), $H_\infty(X_{\sigma,\overline{\sigma}}) \geq k - (t+1)\ell - \log(1/\zeta)$.

Thus, conditioned on any fixed values $\sigma, \overline{\sigma} \in T_{\alpha,\zeta}$, Arthur not outputting $\perp$ in initial sampling and Arthur's coin $b = 0$, Merlin's view is simply

$$D_0^{\sigma,\overline{\sigma}} \equiv (\sigma, Y), \mathcal{U}_m, \overline{\mathrm{2NMExt}}(\overline{\tau}_{L,R}^{\sigma,\overline{\sigma}}((\sigma, Y), X_{\sigma,\overline{\sigma}})).$$

On the other hand, if Arthur's coin is $b = 1$ (and other conditions also hold), Merlin's view is

$$D_1^{\sigma,\overline{\sigma}} \equiv (\sigma, Y), \mathrm{2NMExt}(\sigma, Y, X_{\sigma,\overline{\sigma}}), \overline{\mathrm{2NMExt}}(\overline{\tau}_{L,R}^{\sigma,\overline{\sigma}}((\sigma, Y), X_{\sigma,\overline{\sigma}})).$$

Because 2NMExt is a strong relaxed two-source non-malleable extractor with error δ for independent sources where the left has min entropy at least $n-\ell$ and the right has min entropy at least $k-(t+1)\ell+\log(1/\zeta)$, we have that for any (worst-case) choice of $\sigma, \overline{\sigma} \in T$,

$$D_0^{\sigma,\tilde{\sigma}} \approx_\delta D_1^{\sigma,\tilde{\sigma}}.$$

From the fact that $\Pr[(\Sigma,\overline{\Sigma}) \in T_{\alpha,\zeta}] \geq 1-\zeta-2\gamma/\alpha$, it follows that Merlin's views are at most $\zeta+\delta$ distinguishable:

$$\begin{aligned}
\Delta(D_0^{\Sigma,\overline{\Sigma}}; D_1^{\Sigma,\overline{\Sigma}}) &= \sum_{\sigma,\overline{\sigma}} \Pr[(\Sigma,\overline{\Sigma}) = (\sigma,\overline{\sigma})]\Delta(D_0^{\sigma,\overline{\sigma}}; D_1^{\sigma,\overline{\sigma}}) \\
&= \sum_{\sigma,\overline{\sigma} \notin T_{\alpha,\zeta}} \Pr[(\Sigma,\overline{\Sigma}) = (\sigma,\overline{\sigma})]\Delta(D_0^{\sigma,\overline{\sigma}}; D_1^{\sigma,\overline{\sigma}}) + \sum_{\sigma,\overline{\sigma} \in T_{\alpha,\zeta}} \Pr[(\Sigma,\overline{\Sigma}) = (\sigma,\overline{\sigma})]\Delta(D_0^{\sigma,\overline{\sigma}}; D_1^{\sigma,\overline{\sigma}}) \\
&\leq \zeta + 2\gamma/\alpha + \sum_{(\sigma,\overline{\sigma}) \in T_\zeta} \Pr[(\Sigma,\overline{\Sigma}) = (\sigma,\overline{\sigma})]\delta \\
&\leq \zeta + \delta + 2\gamma/\alpha
\end{aligned}$$

Thus, by Proposition 3 there exists a set Π_N^c such that $\Pr_{(\sigma,y) \xleftarrow{u} \{0,1\}^n}[(\sigma,y) \in \Pi_N^c] \geq 1-1/c$ and for any $(\sigma,y) \in \Pi_N^c$, Merlin's views are at most $c(\delta+\zeta+2\gamma/\alpha)$ distinguishable.

It follows from Proposition 5.1 that for any strategy of Merlin and any input $(\sigma,y) \in \Pi_N$, $\Pr[b'=b] \leq \frac{1+c(\delta+\zeta+2\gamma/\alpha)}{2}$.

Finally, we need to handle the probability that Arthur fails to output $\perp$ (sampling succeeds) for any σ such that $\exists y, (\sigma,y) \in \Pi_N^c$. Recall that any $\sigma \notin S_\alpha$. Thus,

$$\begin{aligned}
\Pr[\text{sampling succeeds}] &= \Pr_{(x,\phi)) \leftarrow C(\mathcal{U})}[x = 1 \wedge \mathsf{Ext}_{\text{samp}}(x) = \sigma] \\
&= \Pr_{x,\phi}[\mathsf{Ext}_{\text{samp}}(x) = \sigma | \phi = 1] \Pr_{\phi}[\phi = 1] \\
&\leq \frac{1+\alpha}{2^\ell} \cdot p
\end{aligned}$$

So finally, we can conclude:

$$\begin{aligned}
\forall (\sigma,y) \in \Pi_N^c,\ \Pr[\text{Arthur accepts } (\sigma,y)] &\leq \Pr[\text{sampling succeeds}] \frac{1+c(\delta+\zeta+2\gamma/\alpha)}{2} \\
&\leq \frac{p(1+\alpha)(1+c(\delta+\zeta+2\gamma/\alpha)}{2 \cdot 2^\ell}.
\end{aligned}$$

We conclude from Claim 5.2 and Claim 5.3, that for any $c > 1$, $\alpha > 0$ $\beta \in (0,1)$, and $\zeta \in (0,1)$ there is a two-round IP protocol where Arthur can be represented by NP-circuit of size $\mathsf{poly}(s(n))$ that recognizes $\Pi = (\Pi_Y^\beta, \Pi_N^c)$ with completeness/soundness gap

$$\frac{(1-\alpha)(1+\beta)}{(1+\alpha)(1+c(\delta+\zeta+2\gamma/\alpha))}$$

Finally, we repeat the base proof system m times in parallel and have Arthur accept if *all* m iterations accept. This results in the following completeness/soundness gap:

$$\left(\frac{(1-\alpha)(1+\beta)}{(1+\alpha)(1+c(\delta+\zeta+2\gamma/\alpha))}\right)^m$$

If we take $\alpha = \frac{\beta/4}{2-\beta/4}$, then we have:

$$\frac{1-\alpha}{1+\alpha} = 1-\beta/4$$

. We take $1/c = \beta = \epsilon/6$. We take $\delta, \zeta, 2\gamma/\alpha = \frac{\beta^2}{9} \leq \frac{1}{3} \cdot \frac{\beta^3}{2\beta+\beta^2}$. These imply,

$$\begin{aligned}\frac{1+\beta}{1+c(\delta+\zeta+2\gamma/\alpha)} &\geq \frac{1+\beta}{1+1/\beta \cdot \frac{\beta^3/2}{\beta+\beta^2/2}} \\ &= \frac{(1+\beta)(\beta+\beta^2/2)}{\beta+\beta^2} \\ &= \frac{(1+\beta)\beta(1+\beta/2)}{\beta(1+\beta)} = 1+\beta/2\end{aligned}$$

So, if we set $m > \log(B^2)/\log(1+\beta/8)$:

$$\begin{aligned}\left(\frac{(1-\alpha)(1+\beta)}{(1+\alpha)(1+c(\delta+\zeta+2\gamma/\alpha))}\right)^m &\geq ((1-\beta/4)(1+\beta/2))^m \\ &= \left(1+\beta/4-\beta^2/8\right)^m \\ &> (1+\beta/8)^{\log(B^2)/\log(1+\beta/8)} = B^2\end{aligned}$$

Thus by Theorem 12, this implies the existence of an $s'(n)$-size nondeterministic NP circuit, $\mathcal{C}$, (where $s'(n) = \mathsf{poly}(s(n)) \geq s(n)$) that decides the promise problem, Π. Because $\Pi_Y^{\epsilon/6}$ is $(\epsilon - \gamma - 2\gamma/\alpha - \beta)$-dense under G (i.e. $\Pr_s[\mathsf{G}(s) \in \Pi_Y^{\epsilon/6}] \geq \epsilon - \gamma - 2\gamma/\alpha - \beta$) and $\Pi_N^{6/\epsilon}$ is $1 - 1/c$-dense under the uniform distribution (i.e. $\Pr_z[z \notin \Pi_N^{6/\epsilon}] \leq 1/c$), the nondeterministic NP circuit $\mathcal{C}$ can distinguish with advantage at least (by our assumption that $\gamma \leq \epsilon/6$)

$$|\epsilon - \gamma - 2\gamma/\alpha - \beta - 1/c| \geq |\epsilon - \epsilon/18 - \epsilon/18 - \epsilon/6 - \epsilon/6| \geq \epsilon/2 \geq 1/s'(n).$$

The first inequality follows from our setting that $\beta = 1/c = \epsilon/6$ and $2\epsilon/\alpha < \beta/3 = \epsilon/18$ and the last follows from the fact that $2/\epsilon = 2s(n) \leq s'(n)$. In conclusion, our initial assumption towards contradiction must be false.

5.3 Removing the No-Fixed Points Assumption

Theorem 13. *Define $\mathcal{X}[k, s(n)]$ be the family of k-min-entropy sources on $\{0,1\}^n$ that are samplable by the post-selection with circuits in $\mathsf{SIZE}[s(n)]$. Assume* NMExt $: \{0,1\}^n \to \{0,1\}^m$ *is a relaxed, seedless ϵ-non-malleable extractor with respect to sources in $\mathcal{X}[k, s_s(n)]$ and tampering functions in $\mathsf{SIZE}[s_t(n)]$. Then* NMExt *is an ϵ'-seedless non-malleable extractor with respect to sources in $\mathcal{X}[k', s'_s(n)]$ and tampering functions in $\mathsf{SIZE}[s_t(n)]$, where*

- $k' := k + t\log(1/\epsilon) + 1$
- $s'_s(n) := s_s(n) - s_t(n) - c \cdot t \cdot n$ *for some constant c.*
- $\epsilon' := 2\epsilon$.

Proof. Let $f_1, \ldots, f_t \in \mathsf{SIZE}^n[s_t(n)]$ and $X \in \mathcal{X}[k', s'_s(n)] \subseteq \mathcal{X}[k, s_s(n)]$.

We begin by partitioning X according to the fixed point patterns induced by $f_1, \ldots, f_t$. For $z \in \{0,1\}^t$ define $S_z = \{x : f_i(x) = x \iff z_i = 1\}$, $\alpha_z := \Pr[X \in S_z]$, and $X_z = X | X \in S_z$. Note that any X_z can be sampled with post-selecting circuits of size $s_s + s_t + cnt$ for some constant c.

Note that $X = \sum_z \alpha_z X_z$.

We now define a set of "good" values z, $G := \{z : \alpha_z \geq \epsilon/2^t\}$ and $S_G := \bigcup_{z \in G} S_z$. Note that for $z \in G$, $H_\infty(X_z) \geq k - t\log(1/\epsilon)$.

Note also that $\Pr[X \notin S_G] = \sum_{z \notin S_G} \alpha_z < 2^t \cdot \epsilon/2^t = \epsilon$.

Hence, for $X' := (X | X \in S_G)$, $X' \approx_\epsilon X$. We can write $X' = \frac{1}{\eta}\sum_{z \in G} \alpha_z X_z$ for a normalization factor $\eta = \sum_{z \in G} \alpha_z$.

For any $z \in G$, let $D_{\overline{f},z}$ be the distribution for each $i \in [t]$ outputs `same` if $z_i = 1$ and $\mathrm{NMExt}(f_i(X_z))$ otherwise. Let $D_{\overline{f}}$ be the distribution that samples $z \in G$ with probability α_z/η, and outputs a sample from $D_{\overline{f},z}$. Then,

$$\begin{aligned}
&\Delta(\mathrm{NMExt}(X), \mathrm{NMExt}(f_1(X)), \ldots, \mathrm{NMExt}(f_t(X)); \mathcal{U}_m, \mathrm{Copy}(D_f, \mathcal{U}_m)) \\
&\quad \leq \epsilon + \Delta(\mathrm{NMExt}(X'), \mathrm{NMExt}(f_1(X')), \ldots, \mathrm{NMExt}(f_t(X')); \mathcal{U}_m, \mathrm{Copy}(D_f, \mathcal{U}_m)) \\
&\quad \leq \epsilon + \sum_{z \in G} \frac{\alpha_z}{\eta} \Delta(\mathrm{NMExt}(X_z), \mathrm{NMExt}(f_1(X_z)), \ldots, \mathrm{NMExt}(f_t(X_z)); \mathcal{U}_m, \mathrm{Copy}(D_f, \mathcal{U}_m)) \\
&\quad \leq \epsilon + \sum_{z \in G} \frac{\alpha_z}{\eta} \epsilon \\
&\quad \leq 2\epsilon.
\end{aligned}$$

5.4 Combining the Results

Combining Lemma 1 and Theorem 13, together with the computational extractor for post-selecting sources referenced in Theorem 10, and the strong relaxed two-source non-malleable extractor referenced in Theorem 9, we obtain the following:

Theorem 14. *If* E *requires exponential-size nondeterministic-circuits, then for any polynomial* $s(n)$ *and* $t < \frac{cn}{\log s(n)}$ *(for some constant* $c < 1$ *there exists a construction* $\mathrm{NMExt} : \{0,1\}^n \to \{0,1\}^m$ *of a* t*-non-malleable seedless extractor for* n*-bit sources with* $c' \cdot n$ *min-entropy (for some constant* $c' < 1$*) samplable by size* $s(n)$ *post-selecting circuits, that is resilient to* $\mathsf{SIZE}[s(n)]$*-tampering with error* $1/s(n)$*. Further, the number of extracted bits is* $m \in \Omega(\frac{n \log\log(n)}{\log(n)})$*, and the extractor runs in time* $s'(n) \in \mathsf{poly}(s(n))$*.*

References

1. Leonard M. Adleman. Two theorems on random polynomial time. In *FOCS*, pages 75–83. IEEE Computer Society, 1978.
2. Divesh Aggarwal, Nico Döttling, Jesper Buus Nielsen, Maciej Obremski, and Erick Purwanto. Continuous non-malleable codes in the 8-split-state model. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 531–561. Springer, Cham, May 2019.
3. Benny Applebaum, Sergei Artemenko, Ronen Shaltiel, and Guang Yang. Incompressible functions, relative-error extractors, and the power of nondeterministic reductions. *Comput. Complex.*, 25(2):349–418, 2016.
4. Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.
5. László Babai. Trading group theory for randomness. In *17th ACM STOC*, pages 421–429. ACM Press, May 1985.
6. László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988.
7. Marshall Ball, Dana Dachman-Soled, and Julian Loss. (Nondeterministic) hardness vs. non-malleability. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 148–177. Springer, Cham, August 2022.
8. Marshall Ball, Eli Goldin, Dana Dachman-Soled, and Saachi Mutreja. Extracting randomness from samplable distributions, revisited. In *64th FOCS*, pages 1505–1514. IEEE Computer Society Press, November 2023.
9. Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 299–315. Springer, Berlin, Heidelberg, August 2003.
10. Mihir Bellare, Oded Goldreich, and Erez Petrank. Uniform generation of np-witnesses using an np-oracle. *Inf. Comput.*, 163(2):510–526, 2000.
11. Matteo Campanelli, Chaya Ganesh, Hamidreza Khoshakhlagh, and Janno Siim. Impossibilities in succinct arguments: Black-box extraction and more. In Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne, editors, *AFRICACRYPT 23*, volume 14064 of *LNCS*, pages 465–489. Springer, Cham, July 2023.
12. Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In Daniel Wichs and Yishay Mansour, editors, *48th ACM STOC*, pages 285–298. ACM Press, June 2016.

13. Eshan Chattopadhyay and Xin Li. Non-malleable codes and extractors for small-depth circuits, and affine functions. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 1171–1184. ACM Press, June 2017.
14. Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 440–464. Springer, Berlin, Heidelberg, February 2014.
15. Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity (extended abstract). In *26th FOCS*, pages 429–442. IEEE Computer Society Press, October 1985.
16. Kai-Min Chung, Huijia Lin, Mohammad Mahmoody, and Rafael Pass. On the power of nonuniformity in proofs of security. In Robert D. Kleinberg, editor, *ITCS 2013*, pages 389–400. ACM, January 2013.
17. Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 601–610. ACM Press, May / June 2009.
18. Andrew Drucker. Nondeterministic direct product reductions and the success probability of SAT solvers. In *54th FOCS*, pages 736–745. IEEE Computer Society Press, October 2013.
19. Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In Andrew Chi-Chih Yao, editor, *ICS 2010*, pages 434–452. Tsinghua University Press, January 2010.
20. Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 465–488. Springer, Berlin, Heidelberg, February 2014.
21. Uriel Feige and Carsten Lund. On the hardness of computing the permanent of random matrices. *Comput. Complex.*, 6(2):101–132, 1997.
22. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011.
23. Oded Goldreich and Maya Leshkowitz. On emulating interactive proofs with public coins. In Oded Goldreich, editor, *Computational Complexity and Property Testing - On the Interplay Between Randomness and Computation*, volume 12050 of *Lecture Notes in Computer Science*, pages 178–198. Springer, 2020.
24. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.
25. Oded Goldreich and Avi Wigderson. Derandomization that is rarely wrong from short advice that is typically good. In *RANDOM*, volume 2483 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2002.
26. Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *18th ACM STOC*, pages 59–68. ACM Press, May 1986.
27. Dan Gutfreund, Ronen Shaltiel, and Amnon Ta-Shma. Uniform hardness versus randomness tradeoffs for arthur-merlin games. *Comput. Complex.*, 12(3-4):85–130, 2003.
28. Russell Impagliazzo and Avi Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *29th ACM STOC*, pages 220–229. ACM Press, May 1997.
29. Mark Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.
30. Dimitar Jetchev and Krzysztof Pietrzak. How to fake auxiliary input. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 566–590. Springer, Berlin, Heidelberg, February 2014.
31. Jeff Kinne, Dieter van Melkebeek, and Ronen Shaltiel. Pseudorandom generators, typically-correct derandomization, and circuit lower bounds. *Comput. Complex.*, 21(1):3–61, 2012.
32. Adam R. Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002.
33. Shi Li. Scheduling to minimize total weighted completion time via time-indexed linear programming relaxations. In Chris Umans, editor, *58th FOCS*, pages 283–294. IEEE Computer Society Press, October 2017.
34. Xin Li. Non-malleable extractors, two-source extractors and privacy amplification. In *53rd FOCS*, pages 688–697. IEEE Computer Society Press, October 2012.
35. Xin Li. New independent source extractors with exponential improvement. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 783–792. ACM Press, June 2013.
36. Xin Li. Three-source extractors for polylogarithmic min-entropy. In Venkatesan Guruswami, editor, *56th FOCS*, pages 863–882. IEEE Computer Society Press, October 2015.
37. Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 1144–1156. ACM Press, June 2017.
38. Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. In *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, pages 28:1–28:49, 2019.

39. Xin Li. Two source extractors for asymptotically optimal entropy, and (many) more. In *64th FOCS*, pages 1271–1281. IEEE Computer Society Press, November 2023.
40. Surya Mathialagan, Spencer Peters, and Vinod Vaikuntanathan. Adaptively sound zero-knowledge SNARKs for UP. Cryptology ePrint Archive, Paper 2024/227, 2024. `https://eprint.iacr.org/2024/227`.
41. Peter Bro Miltersen and N. V. Vinodchandran. Derandomizing arthur-merlin games using hitting sets. *Comput. Complex.*, 14(3):256–279, 2005.
42. Moni Naor. On cryptographic assumptions and challenges. In *Annual International Cryptology Conference*, pages 96–109. Springer, 2003.
43. Noam Nisan and Avi Wigderson. Hardness vs. randomness (extended abstract). In *29th FOCS*, pages 2–11. IEEE Computer Society Press, October 1988.
44. Rafail Ostrovsky, Giuseppe Persiano, Daniele Venturi, and Ivan Visconti. Continuously non-malleable codes in the split-state model from minimal assumptions. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 608–639. Springer, Cham, August 2018.
45. Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 334–354. Springer, Berlin, Heidelberg, March 2013.
46. Rafael Pass. Unprovable security of perfect nizk and non-interactive non-malleable commitments, 2017. `https://www.cs.cornell.edu/~rafael/papers/limits2new.pdf`, Last accessed on 2025-2-10.
47. Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE symposium on security and privacy*, pages 459–474. IEEE, 2014.
48. R. Shaltiel and C. Umans. Pseudorandomness for approximate counting and sampling. In *20th Annual IEEE Conference on Computational Complexity (CCC'05)*, pages 212–226, 2005.
49. Ronen Shaltiel. Weak derandomization of weak algorithms: Explicit versions of yao's lemma. *Comput. Complex.*, 20(1):87–143, 2011.
50. Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *J. ACM*, 52(2):172–216, 2005.
51. Ronen Shaltiel and Christopher Umans. Pseudorandomness for approximate counting and sampling. *Comput. Complex.*, 15(4):298–341, 2006.
52. Ronen Shaltiel and Christopher Umans. Low-end uniform hardness versus randomness tradeoffs for AM. *SIAM J. Comput.*, 39(3):1006–1037, 2009.
53. Luca Trevisan and Salil P. Vadhan. Extracting randomness from samplable distributions. In *41st FOCS*, pages 32–42. IEEE Computer Society Press, November 2000.
54. Salil P. Vadhan and Colin Jia Zheng. A uniform min-max theorem with applications in cryptography. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 93–110. Springer, Berlin, Heidelberg, August 2013.
55. Salil Pravin Vadhan. *A study of statistical zero-knowledge proofs*. PhD thesis, Massachusetts Institute of Technology, 1999.
56. Brent Waters and David J. Wu. Adaptively-sound succinct arguments for NP from indistinguishability obfuscation. Cryptology ePrint Archive, Paper 2024/165, 2024. `https://eprint.iacr.org/2024/165`.
57. Brent Waters and David J. Wu. A pure indistinguishability obfuscation approach to adaptively-sound SNARGs for NP. Cryptology ePrint Archive, Paper 2024/933, 2024. `https://eprint.iacr.org/2024/933`.
58. Brent Waters and Mark Zhandry. Adaptive security in SNARGs via iO and lossy functions. Cryptology ePrint Archive, Paper 2024/254, 2024. `https://eprint.iacr.org/2024/254`.